

UNIVERSIDAD DE ALCALÁ
ESCUELA POLITÉCNICA SUPERIOR
DEPARTAMENTO DE ELECTRÓNICA



**SEGUIMIENTO DE MÚLTIPLES OBJETOS EN ENTORNOS
INTERIORES MUY POBLADOS BASADO EN LA COMBINACIÓN DE
MÉTODOS PROBABILÍSTICOS Y DETERMINÍSTICOS**

TESIS DOCTORAL

AUTORA: Marta Marrón Romera

DIRECTORES: Dr. Miguel Ángel Sotelo Vázquez
Dr. Juan Carlos García García

2008

RESUMEN

La presente tesis se encuentra enmarcada dentro del área de la robótica personal y de servicios. Es éste un área de investigación que ha tomado gran relevancia en las últimas dos décadas gracias a los continuos avances de la tecnología y su inserción en la vida diaria de la sociedad moderna.

Dentro de este contexto, en la tesis se propone un nuevo algoritmo para el seguimiento de múltiples objetos (“*multiple target tracking*”, MTT abreviadamente), concebido para su uso en entornos interiores complejos. El proceso de MTT diseñado, proporciona información completa sobre los diferentes objetos detectados en cada momento en el entorno del robot, indicando el número, posición, velocidad, camino recorrido e identidad de los mismos. Esta información es obtenida por el algoritmo de seguimiento a partir de los datos recogidos por el sistema de observación de entrada al sistema.

La solución propuesta cumple todas las especificaciones establecidas por el comportamiento deseado para el seguidor: ha de tener en cuenta la incertidumbre de los modelos de estado y medida de los objetos bajo seguimiento; ha de ser flexible al uso de distintos tipos de sensores ha de poder adaptarse al tipo de información de entrada al algoritmo que proporcione el sistema (visión, ultrasonidos, infrarrojo, radio frecuencia, etc.) que conformen el sistema de observación empleado; debe ser capaz de seguir los diferentes tipos de objetos que el robot pueda encontrar en su movimiento por el entorno, independientemente de la dinámica o la forma de estos objetos; finalmente, tiene que alcanzar el nivel de robustez y fiabilidad que requiere la aplicación de robótica personal en la que se enmarca, en la cual la seguridad del propio robot y de los objetos seguidos (generalmente personas u otros robots) es una especificación básica.

Para poder cumplir todas las especificaciones necesarias, el algoritmo de seguimiento diseñado en esta tesis adopta como mejor solución la combinación de métodos probabilísticos y determinísticos. De este modo, se propone un filtro de partículas como núcleo de estimación del algoritmo de seguimiento, al cual se le incorporan dos procesos de clasificación que actúan, respectivamente, como algoritmo de asociación y filtro de salida. Esta combinación da lugar al “*Filtro de Partículas Extendido con Proceso de Clasificación*” (“*Extended Particle Filter with Clustering Process*”, XPFCL), nombre con el que se identifica el algoritmo propuesto por la autora para el seguimiento de múltiples objetos en entornos interiores muy poblados.

El filtro de partículas permite modelar múltiples estados en una única distribución multimodal; su flexibilidad lo hace idóneo para su aplicación con distintos tipos de modelos de estado y observación. Tales características convierten a esta versión del filtro de Bayes en la más adecuada para realizar el seguimiento de múltiples objetos, con la prestación adicional de poder realizar tal tarea de seguimiento con un coste computacional prácticamente constante. La idea de usar el filtro de partículas como estimador multimodal en aplicaciones de seguimiento ya ha sido propuesta en varios trabajos previos de investigación, pero la falta de robustez del sistema así obtenido ha llevado en todos los casos a descartar estas soluciones. Esta tesis propone y demuestra que la incorporación de una parte determinística en el algoritmo de

seguimiento basado en filtro de partículas añade la robustez que el estimador multimodal requiere.

En el presente documento, se incluye una profunda revisión (sobre algoritmos y resultados) de los trabajos llevados a cabo por la comunidad científica en esta misma línea de investigación. Además, se muestra también un estudio exhaustivo del comportamiento del sistema de seguimiento propuesto en situaciones complejas en términos de robustez, fiabilidad, eficiencia y tiempo de ejecución. Finalmente, se realiza la comparación de la solución diseñada por la autora con dos de los algoritmos más conocidos y usados por la comunidad científica en tareas de seguimiento similares: el “*Filtro de Asociación Conjunta de Datos*” o “*Joint Probabilistic Data Association Filter*”, en su versión continua (JPDAF) y muestreada (SJPDAF). Estas comparativas permiten contrastar y validar la contribución de la presente tesis en esta área de investigación.

ABSTRACT

This thesis is involved in the area of service and personal robotics, a research topic that has gained importance in the last two decades thanks to the evolution of technology and its insertion in everyday life of people.

In this context, the thesis proposes a multiple target tracking (MTT) algorithm appropriate to be used in complex indoor environments. The MTT gives information about number, position, speed, track and identification of the different objects in the environment of the robot, information extracted from the data collected by the input observation system.

The proposed solution fulfils some specifications derived from the desired performance: it has to consider the uncertainty of objects behaviour and sensor models; it has to be flexible in order to process input data coming from different kinds of sensors (vision, sonar, audio, laser, radio, etc., or some kind of fusion among any of them); it must handle different types of objects that can affect the robot's behaviour in its environment, despite of their dynamics and deformable shape; it has to accomplish real time execution regardless of the number of objects and complexity of the scene being tracked; finally, it has to achieve the level of robustness and reliability needed by personal and service robotics, where safety of the surrounding objects (persons and other robots) and the robot itself is a main requirement.

In order to achieve all the fore mentioned specifications, a combination of probabilistic and deterministic algorithms is proposed as the best solution for the MTT. A particle filter is used as estimation kernel of the tracker, and two deterministic clustering proposals are incorporated to it, where they are used as association process and output filtering, respectively, in the tracking task. This combination results in the "*Extended Particle Filter with Clustering Process*", XPFCP, the proposal for multi-tracking applications presented in this thesis.

Particle filters have the capability to model multiple states within a single distribution and have a great flexibility in managing any kind of dynamics and observation models. These characteristics make this version of the Bayes filter the most suitable algorithm for the multi-tracking solution proposed. Therefore, they can accomplish the tracking task pursuit with an almost constant computational cost. The same approach has been tried in some other research works, but the lack of robustness of the final MTT implementation has lead to discarding these solutions. This thesis proposes and demonstrates that a deterministic part in the MTT adds the robustness that the multimodal estimator needs.

Throughout this thesis it can be found a deep revision of previous works (algorithms and results) carried out by the scientific community in the topic of interest. Then, it has been made an exhaustive study of the proposed tracker behaviour in complex tracking tasks in terms of reliability, efficiency, robustness and execution time. Finally, a comparison among the proposed solution and two of the best known and most widely used multi-tracking algorithms in the scientific community (the "*Joint Probabilistic Data Association Filter*" in its continuous –JPDAF– and sampled –SJPDF– versions) is also performed and analyzed in order to validate the thesis contribution in this research area.

AGRADECIMIENTOS

La cantidad de personas que se involucran profesionalmente en el desarrollo de cualquier tesis, y más concretamente en ésta, tan larga, hace difícil la inclusión de todas ellas de manera específica en este apartado. Deseo, sin embargo, agradecer especialmente la ayuda a aquellas personas cuya aportación ha sido determinante en algún punto de la tesis.

En primer lugar tengo que agradecer a mis directores de tesis Miguel Ángel Sotelo Vázquez y Juan Carlos García García por la búsqueda inicial de ideas, y la labor de orientación que han desarrollado a lo largo de estos años.

Por otro lado, quiero destacar la colaboración de otros compañeros doctorandos, algunos de ellos ya doctores, y algunos otros todavía alumnos, en la puesta a punto de algunos algoritmos. Gracias a David Fernández Llorca por los trabajos de visión y clasificación borrosa compartidos, a Maria Cabello Aguilar, Jesús Cerro Jiménez, Francisco Huerta Sánchez y Álvaro Marcos Ramiro por la colaboración en el arduo trabajo de pruebas, y sobre todo a Daniel Pizarro Pérez, por su continua aportación y debate de ideas, y por la puesta a punto del sistema de tracking en el espacio inteligente.

I also have to thank the people at CAS in KTH (mainly to Henrik, Patric and Danica) for the time they spent with me in Stockholm. I learnt some important questions over there related not only with my topic of interest but also with the way of driving and working in a robotics lab and group. It was, professionally and personally, a quite nice time.

Finalmente, quiero dar las gracias públicamente a Manuel Mazo Quintas por el apoyo técnico que me ha dado de forma constante y altruista, si bien este agradecimiento se enmarca tanto en lo profesional como en lo personal.

Quiero agradecer también, de forma general, al grupo de investigación robótica (ROBOSAFE) y al de espacios inteligentes y transporte (GEINTRA) de la UAH las conversaciones e intercambio de opinión, sobre temas diversos relacionados con algoritmos probabilísticos, detección con visión, control, estimación, etc., siempre enriquecedoras.

Y, por supuesto, a otro nivel también técnico, he de agradecer al Departamento de Electrónica de la UAH, la ayuda económica y el empeño institucional, a veces contundente, que finalmente permite que alcancemos el objetivo de acabar la tesis.

A todos vosotros y a aquellos alumnos, compañeros profesores y PAS que, en los muchos días de variado, arduo y gracias a vosotros, también divertido e interesante trabajo, con una sonrisa sincera habéis hecho esta tesis más completa técnica y personalmente hablando, algo más fácil, mucho más agradable, y en definitiva posible: Gracias.

Como todo en la vida es parte de un todo, en lo personal, gracias por la paciencia, la espera, el apoyo, la comprensión, la diversión, la risa y la vida... sobre todo a Emilio, a mi mama, mi papa y mi hermano, y también a Julia y Ángela y a los más íntimos (Montse y Vero) ... ahora sí, ya está, ¡sigamos compartiendo lo que ahora por fin viene también juntos!

A TOD@S: CARPE DIEM...

ÍNDICE DE CONTENIDOS

RESUMEN	I
ABSTRACT	III
AGRADECIMIENTOS	V
ÍNDICE DE CONTENIDOS	IX
ÍNDICE DE FIGURAS	X
ÍNDICE DE TABLAS	XXXIII
GLOSARIO	XXXV
CAPÍTULO 1. INTRODUCCIÓN A LA TESIS	1
1.1. Marco de desarrollo de la tesis	2
1.1.1. De la robótica industrial a la de servicio y personal.....	2

1.1.2. Necesidades de la robótica personal	4
1.1.3. Modelando la incertidumbre.....	6
1.1.4. Robots personales en entornos densamente poblados	8
1.1.5. Sistemas sensoriales para la extracción de la información	11
1.2. Objetivos y especificaciones de la tesis	14
1.2.1. Descripción funcional del sistema de seguimiento objetivo de la tesis	14
1.2.2. Desglose de objetivos y especificaciones implicadas en el objetivo global	15
1.3. Estructura de la tesis	18
CAPÍTULO 2. REVISIÓN DE CONOCIMIENTOS	21
2.1. Técnicas empleadas en la tarea de seguimiento de objetos	23
2.1.1. Métodos de estimación	23
2.1.2. Métodos de asociación.....	35
2.2. Trabajos previos en el área de seguimiento de objetos.....	45
2.2.1. Seguimiento de un objeto	47
2.2.2. Seguimiento de múltiples objetos con múltiples estimadores	54
2.2.3. Seguimiento de múltiples objetos con un único estimador basado en la extensión del vector de estados.....	57
2.2.4. Seguimiento de múltiples objetos con un único estimador multimodal	62
2.3. Recapitulación y conclusiones.....	65
CAPÍTULO 3. DESCRIPCIÓN GENERAL DEL SISTEMA	69
3.1. El “Filtro de Partículas Extendido con Proceso de Clasificación” (XPFCP).....	69
3.2. Descripción general de los algoritmos que conforman el XPFCP	73
3.2.1. Contenido probabilístico de la propuesta. El filtro de partículas como solución óptima de estimación al problema de seguimiento	73
3.2.2. Contenido determinístico de la propuesta. El clasificador como solución óptima de asociación y robustecimiento del algoritmo de seguimiento	77
CAPÍTULO 4. CONTENIDO PROBABILÍSTICO DE LA PROPUESTA	81
4.1. El filtro de partículas básico. El algoritmo SIR (“Sequential Importance Resampling”).....	82
4.1.1. Muestreo de Monte-Carlo.....	82

4.1.2. Primera versión del filtro de partículas: el algoritmo SIS (“ <i>Sequential Importance Sampling</i> ”).....	86
4.1.3. El proceso de “ <i>resampling</i> ”: el algoritmo SIR (“ <i>Sequential Importance Resampling</i> ”).....	88
4.1.4. Diagrama funcional del algoritmo SIR.....	90
4.1.5. La versión más extendida del algoritmo SIR: el algoritmo “ <i>Bootstrap</i> ”.....	93
4.2. Modificaciones al algoritmo SIR.....	95
4.2.1. El algoritmo de selección.....	96
4.2.2. La función de aproximación a la creencia.....	98
4.3. “Filtro de Partículas Extendido” (XPF).....	107
4.3.1. Funcionalidad del XPF.....	108
4.3.2. Puntos débiles del XPF.....	116
4.4. Recapitulación y conclusiones.....	120
CAPÍTULO 5. CONTENIDO DETERMINÍSTICO DE LA PROPUESTA.....	123
5.1. Algoritmos de clasificación.....	124
5.1.1. Nomenclatura de clasificación.....	125
5.1.2. Espacio de clasificación.....	126
5.2. El algoritmo “<i>k-medias secuencial</i>”.....	128
5.2.1. El “ <i>k-medias</i> ” básico.....	128
5.2.2. El “ <i>k-medias extendido</i> ”.....	129
5.2.3. Análisis de los puntos débiles del “ <i>k-medias extendido</i> ”.....	133
5.2.4. Versión del algoritmo desarrollada en la tesis: el “ <i>k-medias secuencial</i> ”.....	137
5.2.5. Proceso de validación de clases.....	143
5.2.6. Resultados comparativos del “ <i>k-medias secuencial</i> ” con proceso de validación respecto al “ <i>k-medias extendido</i> ”.....	151
5.3. El algoritmo “<i>subtractive enlazado</i>”.....	160
5.3.1. El “ <i>subtractive</i> ” básico.....	161
5.3.2. Análisis de los puntos débiles del “ <i>subtractive</i> ” básico.....	166
5.3.3. Versión del algoritmo desarrollada en esta tesis: el “ <i>subtractive enlazado</i> ”.....	170
5.3.4. Proceso de validación de clases.....	174
5.3.5. Resultados comparativos del “ <i>subtractive enlazado</i> ” con proceso de validación respecto al “ <i>subtractive</i> ” básico.....	182

5.4. Resultados comparativos de los clasificadores propuestos: “<i>k-medias secuencial</i>” y “<i>subtractive enlazado</i>”	191
5.4.1. Fiabilidad, tasa de error, y valores de distorsión y separación	191
5.4.2. Flexibilidad, robustez frente a variaciones de los parámetros	197
5.4.3. Tiempo de ejecución.....	198
5.5. Conclusiones y resumen de las aportaciones realizadas en esta área	200
CAPÍTULO 6. DESCRIPCIÓN FINAL DEL SISTEMA	203
6.1. Incorporación del proceso de clasificación al “<i>Filtro de Partículas Extendido</i>” (XPF): el “<i>Filtro de Partículas Extendido con Proceso de Clasificación</i>” (XPFCP)	204
6.1.1. Descripción general del XPFCP	205
6.1.2. Otras consideraciones sobre el XPFCP	216
6.1.3. Sensibilidad y ajuste de los parámetros del XPFCP	221
6.1.4. Resultados comparativos del XPFCP frente al XPF.....	247
6.1.5. Conclusiones sobre los resultados expuestos.....	253
6.2. Incorporación del proceso de clasificación a la salida del XPFCP	254
6.2.1. Descripción general de la clasificación de partículas a la salida del XPFCP	255
6.2.2. Análisis funcional del clasificador de partículas a la salida del XPFCP	258
6.2.3. Conclusiones sobre los resultados expuestos.....	267
6.3. Resultados globales obtenidos con distintos sistemas de observación	268
6.3.1. Resultados finales obtenidos con un sistema de estéreo visión ubicado sobre un robot móvil (propuesta de observación centralizada)	269
6.3.2. Resultados finales obtenidos con un sistema de visión basado en un anillo de cámaras estático ubicado en el entorno de movimiento del robot móvil (propuesta de observación distribuida).....	273
6.4. Conclusiones y resumen de las aportaciones realizadas en esta área	278
CAPÍTULO 7. COMPARACIÓN TEÓRICO-EMPÍRICA DE LA PROPUESTA	281
7.1. Comparación con propuestas que incluyen clasificación de medidas y/o de partículas	282
7.1.1. Clasificación de partículas del “ <i>Mixture Particle Filter</i> ” (MPF) ([Vermaak03])	283
7.1.2. Clasificación de medidas del “ <i>Boosted Particle Filter</i> ” (BPF) ([Okuma04])	285

7.2. Comparación con otras propuestas típicas de la aplicación de seguimiento de múltiples objetos	287
7.2.1. El “Filtro de Asociación Probabilística Conjunta de Datos” (“Joint Probabilistic Data Association Filter”, JPDAF).....	289
7.2.2. Resultados comparativos del JPDAF frente al XPFCP	295
7.2.3. El “JPDAF Muestreado” (“Sampled JPDAF”, S-JPDAF).....	304
7.2.4. Resultados comparativos del S-JPDAF frente al XPFCP	311
7.3. Conclusiones sobre las comparativas expuestas	319
CAPÍTULO 8. CONCLUSIONES Y TRABAJOS FUTUROS A LA TESIS	323
8.1. Conclusiones extraídas del trabajo global expuesto	323
8.2. Recapitulación de las aportaciones realizadas	325
8.3. Líneas de investigación y aplicaciones futuras.....	326
APÉNDICE I: VERSIONES DEL FILTRO DE PARTÍCULAS	329
I.1. Filtros de partículas normalizados o regularizados	329
I.1.1. “Gaussian Particle Filter” (GPF) “Gaussian Sum Particle Filter” (GSPF).....	330
I.1.2. “Regularized Particle Filter” (RPF)	331
I.2. El algoritmo “BraMBLe”	333
I.3. “Rao-Blackwellized Particle Filter” (RBPF)	334
I.4. “Unscented Particle Filter” (UPF)	336
APÉNDICE II: CONSIDERACIONES SOBRE LOS PROCESOS DE CLASIFICACIÓN	339
II.1. Características de los algoritmos de clasificación.....	340
II.2. Tipos de algoritmos de clasificación.....	341
II.3. Formas de la medida de distancia	344
II.4. Descripción de las variables utilizadas para la validación de los distintos algoritmos de clasificación	347
APÉNDICE III: DESCRIPCIÓN DE LOS EXPERIMENTOS	349
III.1. Experimentales generales usados.....	350
III.2. Tipos de errores generalizados	352
BIBLIOGRAFÍA	353

ÍNDICE DE FIGURAS

Figura 1.1. Comparativa de la robótica industrial, de servicio y personal en función de la autonomía del robot y la complejidad del entorno en el que se puede desenvolver. Reproducción de [Schraft94]	4
Figura 1.2. Ejemplo de las formas más habituales de la PDF $p(x)$ en los algoritmos probabilísticos: continua genérica a la izquierda; continua parametrizable ($p(x) = norm(\mu, \sigma)$), propia del KF en el centro; discreta genérica propia del PF a la derecha	7
Figura 1.3. Robots Minerva y Rhino actuando como guías de los museos Smithsonian's National Museum of American History en Washington (USA) y Deutsches Museum en Bonn (Alemania) respectivamente.	9
Figura 1.4. Robots del proyecto Tourbot en distintas situaciones: Lefkos en la Foundation of the Hellenic World, en Atenas (Grecia) y Albert en la feria Heinz Nixdorf Museums-Forum en Paderborn (Alemania).....	9
Figura 1.5. Robots Flo y Pearl interaccionando con personas. Flo en un programa televisivo de la NBC y Pearl en un centro de personas mayores en Longwood (USA).	9
Figura 1.6. Silla de ruedas Maid en movimiento autónomo por la estación de ferrocarril de Ulm (Alemania).	10
Figura 1.7. Imagen de una escena real extraída de uno de los experimentos que se usan para validar la propuesta de MTT en la tesis. Sobre la imagen izquierda se marcan en color verde a modo de ejemplo el set de puntos a usar como vector de medidas	

de posición por el estimador propuesto. A la derecha se muestra la misma imagen sobre la que se ha dibujado la que debe ser la salida del proceso de MTT propuesto: un modelo prismático para cada uno de los objetos seguidos que indique su existencia y posición en el entorno.	12
Figura 2.1. Representación sobre una imagen real de los espacios UV y XYZ usados, respectivamente, en las tareas de seguimiento en el plano imagen y en el entorno de movimiento del robot. En esta tesis se resuelve el problema de seguimiento en el entorno del robot, por lo que la información de posición ha de ser tridimensional y estar, por lo tanto, referenciada al espacio XYZ	22
Figura 2.2. Diagrama de bloques del modelo genérico de un sistema expresado en variables de estado. El modelo, que aparece recuadrado en línea discontinua, se representa de más adelante como $ABCD$	23
Figura 2.3. Diagrama de bloques del modelo CV en variables de estado, con ruidos de medidas \bar{o}_t y de estado \bar{v}_t	25
Figura 2.4. Diagrama de bloques de un estimador completo clásico aplicado al modelo descrito en <2.9> y mostrado en la Figura 2.3. El estimador aparece recuadrado en línea discontinua.	26
Figura 2.5. Ejemplo funcionamiento de un localizador de Markov sobre un mapa de rejilla de un entorno interior. El mapa se ha obtenido mediante un láser (las zonas ocupadas aparecen en azul, y las zonas libres en blanco) y sobre el se ha dibujado la rejilla de probabilidad que representa la posición en que se encuentra un robot en un instante (en gris). La mayor probabilidad se representa en color más oscuro. Los puntos rojos representan las posiciones más probables del robot, siendo el que aparece con un círculo rojo la posición real. Extraído de [Burgard99].	32
Figura 2.6. Ejemplo de uso de un POMDP para localizar un robot en un entorno mapeado mediante un grafo. La localización del móvil en cada instante se muestra en la imagen mediante un círculo y un número. El número es un identificador que informa del valor discreto más probable de la creencia en cada momento (la posición del robot en el grafo), y los trazos rojos representan la transición entre dos de estos valores. Extraído de [López04].	33
Figura 2.7. Ejemplo de funcionamiento de un localizador de Monte-Carlo. Las imágenes representan momentos consecutivos del proceso de localización de un robot dentro de un mapa conocido a priori. Los puntos negros (que se van concentrando según se va incrementando la certeza de la posición del robot) representan a las partículas del PF. Extraído de [Doucet01].	33
Figura 2.8. Ejemplo de la PDF de una función de creencia $p(x_t y_{1:t})$ multimodal continua (trazo rojo) de la que se han cogido una serie de muestras (puntos verdes) que pueden representar el valor más probable de la variable x de distintos modelos, en lugar de valores de distinta probabilidad de la misma.	47
Figura 2.9. Resultados del algoritmo de seguimiento “ <i>Condensation</i> ” ([Isard98a]) ante una secuencia de video de una persona bailando. Las elipses blancas dibujadas en la imagen muestran la evolución de la posición de la cabeza de la bailarina. Se muestra el número de orden de cada imagen y el instante temporal con el que se corresponde.	48

Figura 2.10. Resultados de los trabajos de seguimiento del movimiento de una cabeza humana, extraídos de [Chen02]. Entre paréntesis se muestra el número de orden de cada imagen dentro de la secuencia de video. 50

Figura 2.11. Resultados extraídos de [Torma03] en el seguimiento de un único objeto mediante un RBPF. Los contornos más probables que se obtienen en cada paso de ejecución como salida del seguidor se muestran en negro alrededor del objeto seguido..... 51

Figura 2.12. Resultados de seguimiento del movimiento de una abeja en un enjambre, presentados en [Khan04]. El movimiento real del insecto, extraído manualmente, se representa en verde, mientras que la salida del estimador probabilístico se presenta en azul..... 52

Figura 2.13. Resultados arrojados por el estimador presentado en [Odobez06] para el seguimiento de una persona. En amarillo se muestra la propuesta más probable, en rojo la media de las partículas más probables y en verde la moda de la misma distribución. 53

Figura 2.14. Resultados del seguimiento de una persona arrojados por el estimador presentado en [Kobayashi06] en un instante de tiempo. Las imágenes muestran la captura de la escena obtenida desde tres de las cámaras usadas como sistema de adquisición. El cuadrado azul muestra la posición estimada del objeto en ellas. 54

Figura 2.15. Resultados mostrados en [Pérez02] del seguimiento de un número variable de personas k en una secuencia de imágenes obtenidas con cámara estática. Cada objeto viene marcado con un rectángulo de un color diferente. El número de orden de la imagen en la secuencia completa se indica bajo cada una de ellas. 55

Figura 2.16. Resultados obtenidos con el algoritmo propuesto en [Tweed02] en distintas imágenes de la secuencia de prueba. La estructura física de los pájaros cuya posición es seguida por el sistema de tracking propuesto se muestra con trazo blanco. 56

Figura 2.17. Ejemplo del modelo de asociación utilizado en [Khan03] para identificar a los objetos de una escena. 56

Figura 2.18. Resultados obtenidos mediante el algoritmo de MTT presentado en [Khan03]. Los cuadrados verdes marcan la posición y orientación del movimiento (marcado por el triángulo verde interior) de las dos hormigas que interactúan y son objeto de seguimiento en distintos momentos del experimento. 57

Figura 2.19. Resultados obtenidos en el seguimiento de la posición de dos personas en un entorno estático mediante el uso del algoritmo expuesto en [MacCormick99]. El algoritmo permite realizar el tracking de las dos personas incluso ante situaciones de oclusión parcial y total de una de ellas..... 58

Figura 2.20. Resultados expuestos en [Tao99] en la aplicación del algoritmo presentado al seguimiento de un número múltiple y variable de objetos k . Los puntos blancos de cada imagen en la secuencia completa representan a las partículas del estimador bayesiano. 59

Figura 2.21. Resultados obtenidos con el algoritmo presentado en [Spengler03] en el seguimiento de dos personas. Las elipses blancas muestran la posición en la

imagen del objeto seguido y las líneas blancas indican el estado del modelo de interrelación usado.	59
Figura 2.22. Resultados del sistema de seguimiento planteado en [Isard01]. Los polígonos de color indican la estimación de posición arrojada por el algoritmo en diferentes instantes de la secuencia de prueba global.	60
Figura 2.23. Resultados extraídos de [Branson05], en el seguimiento de dos ratones. Se muestran ocho imágenes de la prueba (de la a) a la h)) en las que el modelo de contorno asociado a cada ratón se muestra en un color diferente.....	61
Figura 2.24. Resultados extraídos de [Hue01], en la aplicación del algoritmo ahí descrito al seguimiento de tres personas. Se muestran tres imágenes de la secuencia de video de prueba.	62
Figura 2.25. Resultados obtenidos en la aplicación del algoritmo propuesto en [Smith05]. La estimación de posición arrojada por el filtro para cada persona se marca con un rectángulo de color.....	62
Figura 2.26. Resultados del algoritmo de MTT descrito en [Vermaak03]. Cada jugador objeto del seguimiento se marca en un color diferente en las imágenes con tantos rectángulos como partículas asociadas a él haya en la mezcla de componentes.	63
Figura 2.27. Resultados del algoritmo de MTT propuesto en [Okuma04]. Los rectángulos blancos representan a la salida del algoritmo “ <i>Adaboost</i> ”, mientras que los negros representan el valor de las partículas. En estas dos imágenes se aprecia la capacidad del algoritmo de reaccionar correctamente ante la desaparición de objetos.	65
Figura 3.1. Diagrama de bloques general del XPFCP propuesto en la tesis. Los bloques en color naranja representan los pasos básicos del PF y los de color azul los nuevos que se incorporan a éste para dar lugar al algoritmo propuesto. Con línea discontinua se marcan los bloques en los que se usa la información procedente de la clasificación de medidas $Y_t \Rightarrow G_{1:k,t in}$ o de partículas $S_t \Rightarrow G_{1:k,t out}$. Se incluye una imagen obtenida en un experimento real, sobre la que se representa con puntos (coloreados en función de su coordenada x) el set Y_t y la salida determinística del algoritmo caracterizada por un cilindro de color diferente para cada objeto, cuya posición y velocidad es obtenida por el XPFCP en cada instante y conforma su trayectoria en el espacio y el tiempo, con línea punteada y en el mismo color.	70
Figura 3.2. Representación 3D (a la izquierda) y 2D (a la derecha) del valor de las componentes x y z del conjunto de partículas S_t y sus correspondientes pesos \bar{w}_t a la salida del XPFCP, en el instante del experimento real mostrado en la Figura 3.1.	72
Figura 3.3. Multimodalidad de la creencia $p(\bar{x}_t \bar{y}_{1:t})$ generada por el PF en una tarea de seguimiento de múltiples objetos, extraída de [Isard98a]. A la izquierda de la imagen se muestra la evolución temporal de la coordenada u de la PDF $p(u_t \bar{y}_{1:t})$ que modela la posición $\bar{x} = u$ de cada una de las personas en la imagen, que se expone a su vez a la derecha para una de cada cinco ejecuciones del filtro. El instante temporal de proceso de cada imagen se muestra al lado de estas.	76
Figura 4.1. Diagrama funcional del algoritmo PF.	92

Figura 4.2. Representación unidimensional de la evolución de $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$, donde $n = 11$, en las diferentes etapas del PF. El valor de las partículas en el espacio de estado se representa con puntos azules. La función de “importance sampling” $w(x_{0:t})$ usada en el paso de corrección se muestra en naranja, y los puntos azules sobre ella el valor del peso asignado a cada partícula..... 93

Figura 4.3. Representación unidimensional y unimodal del problema asociado a la elección de una función de aproximación a la creencia igual a $p(\bar{x}_t | \bar{x}_{t-1})$, característica del algoritmo “Bootstrap”. En la figura la creencia a priori $p(\bar{x}_t | \bar{y}_{1:t-1})$ representada por $S_{t|t-1}$ se muestra en azul, y la verosimilitud $p(\bar{y}_t | \bar{x}_t)$ se muestra en rojo. 95

Figura 4.4. Representación funcional de un proceso de “movimiento” genérico de $S_t = \{\bar{x}_t^{(i)}, 1/n\}_{i=1}^n$, para un \bar{x}_t unidimensional. El valor en el espacio de estado de las partículas y de su peso se muestra en rojo..... 101

Figura 4.5. Diagrama funcional del algoritmo XPF. Los pasos modificados respecto a la versión básica del PF de la Figura 4.1 se muestran sobre fondo azul. 109

Figura 4.6. Representación unidimensional de la evolución de $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ en las diferentes etapas del XPF. Los pasos modificados respecto a la versión básica del PF de la Figura 4.2 se indican con fondo azul. Las partículas nuevas insertadas en el paso de re-inicialización $\hat{s}_{i,t-1} / i = n - n_m : n$ se muestran en color verde. 110

Figura 4.7. Representación unidimensional del proceso de obtención del valor de $p(y_t | x_t^{(i)})$ usado para calcular el peso de $x_{t|t-1}^{(i)}$, en el paso de corrección del XPF. La línea azul muestra la gaussiana de verosimilitud usada para obtener $\tilde{w}_t^{(i)}$. La media $h(x_{t|t-1}^{(i)}, o_t)$ (representada con línea naranja) y la varianza O (línea negra) de la función la establece el modelo de observación usado, aplicado al valor concreto de la partícula. En el eje de abscisas se muestran con puntos verdes el conjunto Y_t , cuya aplicación a la gaussiana permite obtener $\tilde{w}_t^{(i)}$ 114

Figura 5.1. Representación 3D en el espacio XZY (a la izquierda) y proyección 2D en el plano XZ (a la derecha) de un set de datos Y extraído de un experimento real en el que hay cuatro objetos en una escena. 127

Figura 5.2. Imagen de la escena real de la que se extrae el set de medidas Y representado en la Figura 5.1. La proyección del conjunto en el plano imagen se muestra en verde. 127

Figura 5.3. Diagrama funcional de la versión básica del “k-medias”..... 130

Figura 5.4. Ejemplos de ejecución del “k-medias” básico para un mismo set Y , en un espacio bidimensional ($\lambda_T = 2$), utilizando como parámetro diferenciador la distancia euclídea ($d_{euclídea[i,j]}$), y para un número de clases $k=3$ (imagen izquierda) y $k=2$ (imagen derecha). 130

Figura 5.5. Diagrama funcional del “ <i>k-medias extendido</i> ”. Sus diferencias funcionales respecto al flujograma del “ <i>k-medias</i> ” básico se destacan sobre fondo verde.....	131
Figura 5.6. Ejemplo de ejecución del “ <i>k-medias extendido</i> ” en el experimento descrito en la Figura 5.4. Los círculos azules representan el valor de $distM$ en el espacio de clasificación.	132
Figura 5.7. Ejemplo de ejecución del algoritmo TTSAS en el experimento descrito en la Figura 5.4. El margen de histéresis que se crea en el espacio de características para $distM1 < distM2$ se destaca en azul claro.	133
Figura 5.8. Ejemplos de ejecución del “ <i>k-medias extendido</i> ” en el experimento descrito en la Figura 5.4. Se muestran los resultados del algoritmo para distintos valores del parámetro $distM$: de izquierda a derecha $distM1 = distM0 / 1.5$, $distM0$ y $distM2 = 1.5 \cdot distM0$	135
Figura 5.9. Tasa de error del “ <i>k-medias extendido</i> ” en el experimento complejo en función del valor de $distM$	136
Figura 5.10. Diagrama de bloques del modelo CV modificado (CV) que se emplea en el proceso de predicción de cada centroide $\vec{g}_{j,t t-1}$ de cada clase $G_{j,t} / j=1:k_t$	138
Figura 5.11. Diagrama de bloques del proceso de cálculo del vector de entrada $\vec{u}_{j,t-1}$ del modelo CV (Figura 5.10). Este modelo se aplica a la predicción de valor del centroide de cada clase $\vec{g}_{j,t t-1} / j=1:k_t$, por lo que el ruido de estado \vec{v}_t y medida \vec{o}_t se suponen incluidos en el proceso determinístico de clasificación $Y_t \Rightarrow G_{1:k,t}$	140
Figura 5.12. Diagrama funcional del “ <i>k-medias secuencial</i> ”. Las mejoras aportadas al flujograma del “ <i>k-medias extendido</i> ” de la Figura 5.5 se destacan con fondo verde.....	141
Figura 5.13. Flujograma simplificado del “ <i>k-medias secuencial</i> ” de la Figura 5.12.	142
Figura 5.14. Representación del espacio de definición de las características $d_{j,jold}$ y $p_{j,1:t}$, base del proceso de validación, en función de los correspondientes límites d_{valid} y p_{valid} , y factores de histéresis π_d y π_p	145
Figura 5.15. Representación gráfica de la forma de cálculo del valor límite de la característica de validación de verosimilitud p_{valid} en función del número de clases generadas por el clasificador k_t	146
Figura 5.16. Diagrama explicativo de la funcionalidad general del proceso de validación de clases.	146
Figura 5.17. Diagrama funcional del proceso de validación aplicado al “ <i>k-medias secuencial</i> ”. Las etapas modificadas respecto a la versión genérica (Figura 5.16) se muestran con fondo verde.	148
Figura 5.18. Diagrama funcional del “ <i>k-medias secuencial</i> ” con validación. La figura se obtiene al añadir al flujograma del “ <i>k-medias secuencial</i> ” (Figura 5.13) el proceso de validación de la Figura 5.17 (sobre fondo verde).	148
Figura 5.19. Ejemplo de ejecución del “ <i>k-medias secuencial</i> ” en un espacio bidimensional ($\lambda_T = 2$), frente a los sets consecutivos en el tiempo Y_{t-1} , Y_t , Y_{t+1} e	

Y_{t+2} (imágenes de izquierda a derecha). La evolución muestra una disminución de densidad de datos asociados a la clase G_4 , debido a lo cual ésta tiende confundirse con G_2 149

Figura 5.20. Ejemplo de ejecución del “*k-medias secuencial*” con validación basado únicamente en la condición de distancia, frente a los mismos sets $Y_{t:t+2}$ mostrados en la Figura 5.19. 150

Figura 5.21. Ejemplo de ejecución del “*k-medias secuencial*” con el proceso de validación completo, frente a los mismos sets $Y_{t:t+2}$ mostrados en la Figura 5.19. 150

Figura 5.22. Tasa de error del “*k-medias secuencial*” sin proceso de validación (en tonos rojos) y del “*k-medias extendido*” (en tonos azules) en los experimentos complejo (a) y completo corto (b). 151

Figura 5.23. Tasa de error del “*k-medias secuencial*” sin validación (en tonos rojos) y con validación (en tonos amarillos) en los experimentos complejo (a) y completo corto (b). 152

Figura 5.24. Tasa de error del “*k-medias secuencial*” con proceso de validación en el experimento complejo en función del valor de $distM$ 153

Figura 5.25. Desviación respecto a la media de error del “*k-medias secuencial*” con validación en el experimento complejo para los parámetros: α_{valid} (alfa_valid), $p_{valid|max}$ (p_valid|max), π_p (pi_p) y π_d (pi_d)..... 154

Figura 5.26. Representación por colores de la posición de $\bar{g}_{1,5}$, proyectada en el plano XZ , de cada uno de los objetos identificados por el “*k-medias secuencial*” sin (imagen izquierda) y con validación (imagen de la derecha). Los resultados se han obtenido ejecutando los algoritmos en la segunda parte del experimento complejo. 155

Figura 5.27. Resultados reales comparativos obtenidos del experimento complejo. En las figuras superiores se muestra su proyección sobre la imagen, y en las inferiores sobre el plano del suelo (XZ). Además, en ambas proyecciones se representa en verde la proyección de los datos de Y 156

Figura 5.28. Gráficas comparativas del t_{exe} y el número de iteraciones de los algoritmos “*k-medias extendido*” y “*k-medias secuencial*” con validación, obtenidas en el experimento complejo. 157

Figura 5.29. Gráficas comparativas del t_{exe} (imagen izquierda) y la media de veces que iteran (imagen derecha) los algoritmos “*k-medias secuencial*” con validación y “*k-medias extendido*”, en función del número de clases que arrojan como salida k_t en el experimento completo largo. 158

Figura 5.30. Gráficas comparativas de los valores de separación y distorsión que arrojan los algoritmos “*k-medias secuencial*” y “*k-medias extendido*”, al ser ejecutados en el experimento complejo..... 159

Figura 5.31. Diagrama funcional de la versión básica del “*subtractive*”. 164

Figura 5.32. Ejemplo de funcionamiento del “*subtractive*” básico en un espacio bidimensional ($\lambda_T = 2$). Las imágenes muestran la evolución de la PDF que caracteriza al set Y en tres iteraciones. El máximo de la PDF se muestra en rojo si su valor es superior al límite p_{th} 164

Figura 5.33. Proyección de los resultados mostrados en la Figura 5.32 sobre el espacio bidimensional de clasificación ($\lambda_T = 2$). Con círculo rojo y azul claro se representan, respectivamente, el valor de $distA$ y $distB$ en el espacio de clasificación.	165
Figura 5.34. Tasa de error del “ <i>subtractive</i> ” básico en función del método usado para el cálculo del límite de verosimilitud p_{th} . Los resultados de sensibilidad al ruido se obtienen con test de ruido y el resto con el complejo.....	167
Figura 5.35. Tasa de error total del “ <i>subtractive</i> ” básico en el experimento complejo, en función del método de cálculo del límite p_{th} , para distintos valores del parámetro α_{new} (alfa_new).	167
Figura 5.36. Tasa de error del “ <i>subtractive</i> ” básico en el experimento complejo, en función del valor de $distA$	168
Figura 5.37. Tasa de error del “ <i>subtractive</i> ” básico en el test complejo, en función de $distB$, ajustado $distA$	169
Figura 5.38. Ejemplos de funcionamiento del proceso de enlazado propuesto para el “ <i>subtractive</i> ” básico sobre dos conjuntos de datos. La pertenencia de cada \bar{y}_i a una u otra clase, y la representación del parámetro $distA$, se muestra con colores. La zona de pertenencia borrosa de la imagen izquierda entre G_2 y G_3 aparece en azul claro.	171
Figura 5.39. Gráficas comparativas de los valores de separación y distorsión que arroja el “ <i>subtractive</i> ” en el experimento complejo, con el método de asignación de miembros propuesto (basado en la distancia, en rojo), y con el propuesto en [Everitt01] (basado en la reducción de la verosimilitud, en verde).	172
Figura 5.40. Resultados arrojados por el “ <i>subtractive</i> ” básico con anulación de verosimilitud (imágenes superiores) y sin él (imágenes inferiores), en una escena real. En cada fila se muestra la evolución en tres iteraciones de la PDF normalizada que caracteriza al set de medidas Y . Su máximo se muestra en rojo si es superior al límite p_{th}	173
Figura 5.41. Proyección sobre el plano imagen de los puntos de Y (en verde claro), y de los resultados de clasificación mostrados en la Figura 5.40. La imagen izquierda muestra el resultado correcto, generado por el “ <i>subtractive</i> ” con anulación de verosimilitud, y la de la derecha el generado por el “ <i>subtractive</i> ” básico, que presenta un error de duplicación de una clase.	174
Figura 5.42. Diagrama funcional del “ <i>subtractive enlazado</i> ”. Las mejoras aportadas al flujograma del “ <i>subtractive</i> ” básico se destacan con fondo verde.	175
Figura 5.43. Diagrama funcional del “ <i>subtractive enlazado</i> ” con proceso de identificación (en amarillo). Las diferentes etapas que lo conforman se diferencian con fondo verde. El bloque azul sustituye al flujograma del “ <i>subtractive enlazado</i> ” tal y como aparece en la Figura 5.42.	177
Figura 5.44. Tasa de error del “ <i>subtractive enlazado</i> ” en función del método usado para la obtención del valor de verosimilitud instantánea de la clase $p_{j,t}$, en el experimento complejo.	179

Figura 5.45. Diagrama funcional del proceso de validación genérico aplicado al “*subtractive enlazado*” con proceso de identificación. Las etapas modificadas respecto al flujograma de la versión genérica (Figura 5.16) se muestran con fondo verde. 180

Figura 5.46. Diagrama funcional del “*subtractive enlazado*” con validación. La figura se obtiene al añadir al flujograma del “*subtractive enlazado*” con proceso de identificación (Figura 5.43 simplificada) el proceso de validación de la Figura 5.45 (sobre fondo verde). 180

Figura 5.47. Tasa de error de los algoritmos “*subtractive enlazado*” (en tonos rojos) y “*subtractive*” básico (en tonos verdes) en los experimentos complejo (a) y completo corto (b). 182

Figura 5.48. Tasa de error del algoritmo “*subtractive enlazado*” sin (en tonos rojos) y con proceso de identificación y de validación (en tonos amarillos) en los experimentos complejo (a) y completo corto (b). 184

Figura 5.49. Tasa de error del “*subtractive enlazado*” con procesos de identificación y validación en el experimento complejo, en función del valor de $distA$ 185

Figura 5.50. Tasa de error del “*subtractive enlazado*” con procesos de identificación y validación en el experimento complejo, en función del valor del parámetro α_{new} (alfa_new). 186

Figura 5.51. Desviación respecto a la media de error del “*subtractive enlazado*” con identificación y validación en el test complejo para los parámetros: α_{valid} (alfa_valid), $p_{valid|max}$ (p_valid|max), π_p (pi_p) y π_d (pi_d). 186

Figura 5.52. Representación por colores de la posición de $\vec{g}_{1,4}$, proyectada en el plano XZ, de cada objeto identificados por el “*subtractive enlazado*” con identificación y sin validación (imagen izquierda) / con validación (imagen derecha). Los resultados se obtienen ejecutando los algoritmos en la segunda parte del test complejo. 187

Figura 5.53. Gráfica comparativa del t_{exe} de la versión básica del “*subtractive*” y del “*subtractive enlazado*” con procesos de identificación y validación, en el experimento complejo. 189

Figura 5.54. Gráfica comparativa del t_{exe} de la versión básica del “*subtractive*” y del “*subtractive enlazado*” con procesos de identificación y validación, en función del número de clases k_i que generan, en el test completo largo. 189

Figura 5.55. Gráficas comparativas de los valores de separación y distorsión que arroja el “*subtractive*” en sus distintas versiones, al ser ejecutados en el experimento complejo. 190

Figura 5.56. Tasa de error de los algoritmos “*k-medias secuencial*” con validación (en tonos azules) y “*subtractive enlazado*” con identificación y validación (en tonos verdes) en los experimentos complejo (a) y completo corto (b). 192

Figura 5.57. Tasa de ruido no filtrado de las distintas versiones del “*k-medias*” y del “*subtractive*” en el experimento de ruido. 194

Figura 5.58. Representación por colores de la posición de $\vec{g}_{1,4}$, proyectada en el plano XZ, de cada uno de los objetos identificados por el “*k-medias secuencial*” con

validación (imagen izquierda) y por el “ <i>subtractive enlazado</i> ” con identificación y validación (imagen derecha). Los resultados se han obtenido ejecutando los algoritmos en la segunda parte del experimento complejo. Cerca del final del experimento el objeto asociado a G_3 oculta totalmente al asociado a G_4 , pues este último pasa antes por el punto de cruce entre las dos clases.	195
Figura 5.59. Gráficas de los valores de separación y distorsión que arroja el “ <i>k-medias secuencial</i> ” con validación y el “ <i>subtractive enlazado</i> ” con identificación y validación, al ser ejecutados en el experimento complejo.....	196
Figura 5.60. Gráficas de los valores de separación y distorsión que arroja el “ <i>k-medias secuencial</i> ” con validación y el “ <i>subtractive enlazado</i> ” con identificación y validación, al ser ejecutados en el experimento complejo.....	196
Figura 5.61. Desviación respecto a la media de error del “ <i>subtractive enlazado</i> ” con identificación y validación en el test complejo para los parámetros: α_{valid} (alfa_valid), $p_{valid max}$ (p_valid max), π_p (pi_p) y π_d (pi_d).....	197
Figura 5.62. Grafica comparativa del t_{exe} del “ <i>k-medias secuencial</i> ” con validación y del “ <i>subtractive enlazado</i> ” con identificación y validación, en el experimento complejo.....	199
Figura 5.63. Gráficas comparativas del t_{exe} del “ <i>k-medias secuencial</i> ” con validación y del “ <i>subtractive enlazado</i> ” con identificación y validación, en función del número de clases que arrojan como salida (gráfica izquierda), y de éste y el número de datos a clasificar (gráfica derecha) en el test completo largo.	199
Figura 6.1. Diagrama funcional del algoritmo XPFCP. Los pasos modificados respecto al flujograma del XPF de la Figura 4.5 se muestran sobre fondo azul, así como los que implican una llamada al proceso de clasificación de medidas $G_{j,t-1 in} / j = 1 : k_{in,t-1}$. La llamada al clasificador de partículas $S_t \Rightarrow G_{j,t out} / j = 1 : k_{out,t}$ se muestra en verde.	206
Figura 6.2. Representación unidimensional de la evolución de $S_t = \{\tilde{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ en las diferentes etapas del XPFCP. Los pasos modificados respecto al flujograma del XPF de la Figura 4.6 se indican con fondo azul, y la clasificación de partículas $S_t \Rightarrow G_{j,t out} / j = 1 : k_{out,t}$ con fondo verde. Las partículas nuevas insertadas de las clases de medidas en el paso de re-inicialización $Y_{t-1} \Rightarrow G_{j,t-1 in} / j = 1 : k_{in,t-1}$ se muestran en rombos verdes.	207
Figura 6.3. Representación unidimensional del proceso de obtención de los pesos en el paso de corrección del XPFCP. La línea azul muestra la gaussiana de verosimilitud usada $p(g_i x_t^{(i)})$. El valor de distancia base del cálculo de $\tilde{w}_t^{(i)} / i = 1 : n$ se calcula entre la media $h(x_{t t-1}^{(i)}, o_t)$ de la gaussiana (con línea naranja) y el centroide más próximo $g_{1:k,t in}$ (en naranja igual que la distancia). En rojo se muestran las agrupaciones de medidas $Y_t \Rightarrow G_{1:k,t in}$ (representadas con puntos verdes), cuyo centroide se indica con un rombo del mismo color.	219

Figura 6.4. Tasa de error generada por el XPFCP en función del número de partículas n de éste, al ejecutarlo en el experimento complejo (en azul) y en el completo-corto (en verde), así como la media de ambos resultados (en rojo). 222

Figura 6.5. Gráficas comparativas del tiempo de ejecución t_{exe} del XPFCP en función de n , obtenidas en el experimento complejo. 223

Figura 6.6. Gráficas comparativas del valor medio del t_{exe} del XPFCP (imagen izquierda) y del tiempo de ejecución t_{exe} por partícula (imagen derecha) en función de n , en los mismos experimentos de la Figura 6.4. 223

Figura 6.7. Gráficas comparativas del valor de \hat{n}_{eff} generado por el XPFCP en función de n , obtenidas en el experimento complejo. Las líneas de trazo grueso muestra el valor medio de las de trazo fino. 224

Figura 6.8. Gráficas comparativas del valor medio de \hat{n}_{eff} generado por el XPFCP en función de n , en los en los mismos experimentos de la Figura 6.4. 225

Figura 6.9. Tasa de error generada por el XPFCP en función del valor de $\gamma_t = \frac{n_{m,t-1}}{n}$ (gamma= n_m/n), en el experimento complejo (en azul) y en el completo-corto (en verde), y valor medio de estos resultados (en rojo). 226

Figura 6.10. Gráficas comparativas del valor de \hat{n}_{eff} generado por el XPFCP en función del valor de γ_t (gamma), obtenidas en el experimento complejo. Las líneas de trazo grueso muestran el valor medio de las de trazo fino. 226

Figura 6.11. Gráficas comparativas del valor medio de \hat{n}_{eff} generado por el XPFCP en función del valor de γ_t (gamma), en los en los mismos experimentos de la Figura 6.9. 227

Figura 6.12. Tasa de error generada por el XPFCP en función del valor porcentual del parámetro n_{init} relativo al número total de partículas n usadas ($\% n_{init}/n$, $\%n_{init}/n$), en el experimento complejo (en azul) y en el completo-corto (en verde), y valor medio de estos resultados (en rojo). 228

Figura 6.13. Imágenes extraídas de la evolución del set de partículas en la ejecución en tiempo real de la segunda parte del experimento complejo, del XPFCP sin (imágenes superiores) y con (imágenes inferiores) refuerzo de las hipótesis nuevas. La gráfica izquierda muestra el peso normalizado de cada partícula $\tilde{w}_i^{(1:n)}$ tras el paso de corrección; la central el número de “hijos” que obtiene cada una de ellas tras el de selección; y la derecha la representación de éste último set S_t (que constituye la salida del XPFCP) en la imagen de entrada. En todas las gráficas, los puntos rojos representan la proyección de cada partícula en el espacio UV (gráficas derechas) y en el XZ (gráficas izquierdas y centrales), y los verdes la de las medidas de Y_{t-1} en el instante anterior. Se incluye el tiempo de ejecución y el valor del factor de calidad \hat{n}_{eff} (neff) generado por el XPFCP cada caso en el instante t 229

- Figura 6.14. Gráficas comparativas del valor medio de \hat{n}_{eff} generado por el XPFCP en función del valor porcentual de n_{init} relativo a n ($\%n_{init}/n$, $\%n_{-init}/n$), en los experimentos de la Figura 6.12.231
- Figura 6.15. Representación gráfica de relación lineal limitada entre el número total de partículas a insertar en la re-inicialización del XPFCP $n_{m,t-1}$ y el de clases de medidas nuevas detectadas por el segmentador en Y_{t-1}232
- Figura 6.16. Tasa de error generada por el XPFCP en los experimentos complejo y completo-corto, en función de la aplicación de ponderación a los pesos \bar{w}_t (con ponderación en azul oscuro y sin ponderación en azul claro), por la verosimilitud $p_{1:k,1:t|in}$ de la clase de medidas $G_{j,t|in} / j = 1:k_{in,t}$ a la que se asocia cada una.233
- Figura 6.17. Gráficas comparativas del valor de \hat{n}_{eff} generado por el XPFCP en el experimento complejo en función de la ponderación o no del array de pesos \bar{w}_t por la verosimilitud $p_{1:k,1:t|in}$ de la clase de medidas $G_{j,t|in} / j = 1:k_{in,t}$ correspondiente. Las líneas de trazo grueso muestran el valor medio de las de trazo fino. Se incluye además una tabla con el valor medio de \hat{n}_{eff} en ese test (a) y en el completo-corto (b).233
- Figura 6.18. Imágenes extraídas de la evolución del set de partículas en la ejecución en tiempo real del XPFCP con proceso de ponderación de pesos, en la segunda parte del experimento complejo. El significado de las gráficas incluidas en la figura es el mismo que el de las incluidas en la Figura 6.13.234
- Figura 6.19. Tasa de error generada por el XPFCP en función de la desviación típica ($\sigma_{v,i}$, sigma-vi) de las componentes del vector de ruido de estado \vec{v}_t , supuestas gaussianas, incorreladas y de media nula, en el experimento complejo (en azul) y en el completo-corto (en verde), y valor medio de ambos resultados (en rojo).237
- Figura 6.20. Gráficas comparativas del valor de \hat{n}_{eff} generado por el XPFCP en el experimento complejo, y en función de la desviación típica ($\sigma_{v,i}$, sigma-vi) de las componentes del vector de ruido de estado \vec{v}_t , supuestas gaussianas, incorreladas y de media nula. Las líneas de trazo grueso muestran el valor medio de las de trazo fino. Se incluye además una tabla con el valor medio de \hat{n}_{eff} en ese test (a) y en el completo corto (b).237
- Figura 6.21. Tasa de error generada por el XPFCP en función de la desviación típica ($\sigma_{o,i}$, sigma-oi) de las componentes del vector de ruido de medida \vec{o}_t supuestas gaussianas, incorreladas y de media nula, en los mismos experimentos de la Figura 6.19.238
- Figura 6.22. Gráficas comparativas del valor de \hat{n}_{eff} generado por el XPFCP en el experimento complejo, y en función de la desviación típica ($\sigma_{o,i}$, sigma-oi) de las componentes del vector de ruido de medida \vec{o}_t , supuestas gaussianas, incorreladas y de media nula. Las líneas de trazo grueso muestran el valor medio de las de trazo

finó. Se incluye una tabla con el valor medio de \hat{n}_{eff} en ese test (a) y en el completo corto (b)..... 239

Figura 6.23. Tasa de error generada por el XPFCP en función del algoritmo de resampling empleado (multinomial en azul intenso, residual en azul oscuro y sistemático en azul claro), en los experimentos complejo y completo-corto..... 240

Figura 6.24. Tasa de error generada por el XPFCP en función del algoritmo de clasificación de medidas usado (“*k-medias secuencial*” con validación en azul, o “*subtractive enlazado*” con identificación y validación en verde), en el experimento complejo y en el completo-corto..... 242

Figura 6.25. Gráficas comparativas del tiempo de ejecución t_{exe} del XPFCP en función del algoritmo de clasificación de medidas usado (con “*k-medias secuencial*” con validación en azul oscuro, y el del clasificador en azul claro, o con “*subtractive enlazado*” con identificación y validación en verde oscuro y el de este clasificador en verde claro), obtenidas en el experimento complejo..... 242

Figura 6.26. Gráficas comparativas del valor de \hat{n}_{eff} generado por el XPFCP en función del clasificador de medidas usado y en los experimentos de la Figura 6.25. Las líneas de trazo grueso muestran el valor medio de las de trazo fino. Se incluye una tabla con el valor medio de \hat{n}_{eff} en ese test (a) y en el completo corto (b). 243

Figura 6.27. Tasa de error obtenida a la salida del XPFCP en función del valor asignado al parámetro β_{valid} (beta_valid), en el experimento complejo..... 245

Figura 6.28. Tasa de error obtenida a la salida del XPFCP en función del valor asignado a los parámetros $p_{valid|max}$ (p_valid|max) y α_{valid} (alfa_valid), en el experimento complejo. 247

Figura 6.29. Tasa de error generada por el XPF, el XPF con las modificaciones del paso de re-inicialización propuestas en la tesis, y el XPFCP, en los experimentos complejo, completo-corto y de ruido. 248

Figura 6.30. Imágenes extraídas de la ejecución en el experimento complejo en tiempo real del XPF (columna izquierda de imágenes), el XPF con las modificaciones del paso de re-inicialización propuestas en la tesis, (columna central) y el XPFCP (columna derecha). La fila superior de imágenes es una representación discreta de la creencia representada por el set $S'_t = \{x_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ a la salida del paso de corrección, y la fila inferior la representación del set S_t de salida de los filtros en la imagen de entrada a éstos. En estas últimas gráficas los puntos rojos representan la proyección de la posición caracterizada por cada partícula de S_t en el espacio UV , y los verdes la de las medidas de Y_t en el mismo espacio. Además se incluye el tiempo de ejecución y el valor del factor de calidad \hat{n}_{eff} (neff) generado por cada uno de los algoritmos en la iteración mostrada..... 250

Figura 6.31. Gráficas comparativas del valor de \hat{n}_{eff} generado por en el experimento complejo por el XPF, el XPF con las modificaciones al paso de re-inicialización propuestas, y el XPFCP. Las líneas gruesas muestran el valor medio de las de trazo

fino. Se incluye una tabla con el valor medio de \hat{n}_{eff} en ese test (a) y en el completo-corto (b).....	251
Figura 6.32. Gráficas comparativas del número de “hijos” obtenido para cada partícula insertada en el paso de re-inicialización $\hat{s}_{i,t-1} / i = n - n_{m,t-1} : n$ relativo al número de partículas $n_{m,t-1}$ insertadas en S_{t-1} en esa etapa, para los mismos algoritmos y en las mismas condiciones de la Figura 6.31.	251
Figura 6.33. Ejemplo de representación en el espacio XYZ del resultado que genera el clasificación de partículas $S_t \Rightarrow G_{1:k,t,out}$, incluyendo con distintos colores el cilindro que caracteriza la estimación de posición de cada objeto relacionado con cada clase de salida (con el valor numérico de su centroide en m en el espacio XZ), y una flecha de color más claro indicadora de la magnitud (como movimiento del centroide en 1s) y dirección de su velocidad (con el valor numérico de ésta en m/s en el espacio $vXvZ$).	257
Figura 6.34. Ejemplo de representación en el espacio XZ (dibujo izquierdo) y en la imagen obtenida con el sistema de sensado (dibujo derecho), de los resultados presentados en la Figura 6.33. Además de la proyección de los datos mostrados en la Figura 6.33 sobre estos espacios, la representación en el plano XZ (dibujo izquierdo) incluye sendos círculos indicadores de la incertidumbre de velocidad en un t_s del objeto correspondiente.....	258
Figura 6.35. Tasa de error generada por el XPFCP en función del algoritmo de clasificación de medidas usado (“ <i>k-medias secuencial</i> ” con validación en azul, o “ <i>subtractive enlazado</i> ” con identificación y validación en verde), en el experimento complejo (tonos claros) y en el completo-corto (tonos oscuros).	259
Figura 6.36. Proyección en la imagen de entrada de un experimento real de tracking extraído del test complejo, de los resultados obtenidos al clasificar el set de partículas S_t generado por el XPFCP con un “ <i>k-medias secuencial</i> ” con validación (imagen izquierda), y con un “ <i>subtractive enlazado</i> ” con identificación y validación (imagen derecha). El formato y la información incluida es la de la Figura 6.34, a excepción de la flecha indicativa de la velocidad de cada objeto, eliminada para clarificar la representación.....	261
Figura 6.37. Representación por colores de la posición de $\bar{g}_{1:4,out}$, proyectada en el plano XZ , de cada uno de los objetos identificados por el “ <i>k-medias secuencial</i> ” con validación (imagen izquierda) y por el “ <i>subtractive enlazado</i> ” con identificación y validación (imagen derecha) en el set de partículas generado a la salida del XPFCP. Los resultados se han obtenido en la segunda parte del experimento complejo. Al final del experimento el objeto asociado a $G_{4,out}$ oculta totalmente al asociado a $G_{3,out}$, pues pasa antes por el punto de cruce de estas clases.	262
Figura 6.38. Gráficas comparativas del tiempo de ejecución t_{exe} del XPFCP en el experimento complejo, en función del algoritmo de clasificación de partículas usado (con “ <i>k-medias secuencial</i> ” con validación en azul oscuro, y el del clasificador en azul claro, o con “ <i>subtractive enlazado</i> ” con identificación y validación en verde oscuro y el de este clasificador en verde claro).	263

Figura 6.39. Tasa de error obtenida a la salida del clasificador del set de partículas generado por el XPFCP $S_t \Rightarrow G_{1:k,t|out}$, en función del valor del parámetro β_{valid} (beta_valid), en el experimento complejo..... 266

Figura 6.40. Tasa de error obtenida a la salida del clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$, en el test complejo, en función del valor de los parámetros $p_{valid|max}$ (p_valid|max), α_{valid} (alfa_valid), π_p (pi_p) y π_d (pi_d)..... 267

Figura 6.41. Secuencia de pares de imágenes (organizadas en columna y ordenados de izquierda a derecha y de arriba a abajo) extraídas de la ejecución en tiempo real del XPFCP en la propuesta de observación centralizada (experimento completo largo). Cada par de imágenes incluye la proyección de los resultados obtenidos en dos planos: en el plano imagen en la imagen superior, y en el plano XZ en la imagen inferior. El formato y la información incluida es la de la Figura 6.34. 272

Figura 6.42. Imágenes de una escena real extraídas con el sistema de observación distribuido usado para validar el comportamiento del XPFCP. Las imágenes de la izquierda muestran la escena tal y como se obtiene de dos de las cuatro cámaras de este sistema de observación. La imagen de la derecha muestra el conjunto de blobs obtenidos a la salida de este sistema de observación, y que constituyen el array de medidas de entrada al XPFCP. 273

Figura 6.43. Gráfica comparativa del numero de objetos seguido por el XPFCP en la propuesta de observación distribuida (kXPFCP, gráfica azul), frente al observado mediante “ground truth” manual (kReal, en verde) La diferencia entre estos dos valores se muestra también en la figura en rojo (Error=kReal-kXPFCP). 276

Figura 6.44. Secuencia de tríos de imágenes (organizadas en fila y ordenados de arriba a abajo) extraídas de la ejecución en tiempo real del XPFCP en la propuesta de observación distribuida. Cada fila de imágenes incluye la proyección de los resultados obtenidos en tres planos (de izquierda a derecha): en el plano imagen de dos de las cuatro imágenes de entrada al filtro, y en el plano XZ. El formato y la información incluida es la de la Figura 6.41, excluyendo la información de velocidad estimada para cada objeto por claridad de la imagen. 277

Figura 7.1. Diagrama funcional del algoritmo JPDAF aplicado a la tarea de MTT de interés. Las etapas del núcleo de estimación (los KFs) se muestran en color naranja, la del algoritmo de asociación JPDA en malva, y los pasos del proceso de modificación del número de hipótesis, basado en el gating, en verde. 296

Figura 7.2. Tasa de error generada por el JPDAF en función de la desviación típica ($\sigma_{v,i}$, sigma-vi) de las componentes del vector de ruido de estado \bar{v}_t , supuestas gaussianas, incorreladas y de media nula, en el experimento complejo (en azul) y en el completo corto (en verde), y valor medio de ambos resultados (en rojo). Esta figura es equivalente a la Figura 6.19 del capítulo 6, que muestra la sensibilidad del XPFCP en estas condiciones..... 297

Figura 7.3. Tasa de error generada por el JPDAF en función de la desviación típica ($\sigma_{o,i}$, sigma-oi) de las componentes del ruido de medida \bar{o}_t , en los tests mostrados en la Figura 7.2. Esta figura es equivalente a la Figura 6.21 del capítulo 6, que muestra la sensibilidad del XPFCP en estas condiciones..... 297

Figura 7.4. Tasa de error generada por el JPDAF en función del valor asignado al parámetro de gating $distG$ en el experimento complejo. Esta figura es equivalente a la Figura 5.23 del capítulo 5 que muestra la sensibilidad, en estas mismas condiciones, del clasificador que actúa como proceso de asociación en el XPFCP.....	299
Figura 7.5. Tasas de error generadas por el JPDAF (en verde) y el XPFCP (en azul) en función del tipo de error, en los tests complejo (tonos claros), completo corto (tonos oscuros) y de ruido (para la tasa de ruido no filtrado).....	300
Figura 7.6. Representación por colores de la posición en el plano XZ estimada en cada instante por el XPFCP (marcada como $\bar{g}_{j out}$ en la imagen de la izquierda) y por el JPDAF (marcada como \bar{x}_j en imagen de la derecha) de los cuatro objetos implicados en el proceso de tracking en la segunda parte del experimento complejo.....	301
Figura 7.7. Gráficas comparativas del t_{exe} del XPFCP (en azul) y del JPDAF (en verde) en el test complejo.....	302
Figura 7.8. Gráficas comparativas del t_{exe} del XPFCP (en azul) y del JPDAF (en verde) en el test completo largo.....	303
Figura 7.9. Diagrama funcional del algoritmo S-JPDAF aplicado a la tarea de MTT de interés. Las etapas del núcleo de estimación (los PFs) se muestran en color naranja, la del algoritmo de asociación JPDA en malva, y los pasos del proceso de modificación del número de hipótesis, basado en el gating, en verde.....	310
Figura 7.10. Tasa de error generada por el S-JPDAF en función de la desviación típica ($\sigma_{o,i}$, sigma-oi) de las componentes del vector de ruido de medida \bar{o}_t , supuestas gaussianas, incorreladas y de media nula, en el test complejo (en azul) y en el completo corto (en verde), y valor medio de ambos resultados (en rojo). Esta figura es equivalente a las Figuras 6.21 del capítulo 6 y 7.3, que muestran la sensibilidad del XPFCP y del JPDAF, respectivamente, en estas condiciones.....	313
Figura 7.11. Tasa de error generada por el S-JPDAF en función de la desviación típica ($\sigma_{v,i}$, sigma-oi) de las componentes del ruido de estado \bar{v}_t en las condiciones de test de la Figura 7.10. Esta figura es equivalente a las 6.19 del capítulo 6 y Figura 7.2 al mostrar la sensibilidad del XPFCP y del JPDAF, respectivamente.....	314
Figura 7.12. Tasa de error generada por el S-JPDAF en función del número de partículas n de cada PF incluido en el seguidor, al ejecutarlo en el experimento complejo (en azul) y en el completo corto (en verde), así como la media de ambos resultados (en rojo). Esta figura es equivalente a Figura 6.4 del capítulo 6 obtenida para el XPFCP.	314
Figura 7.13. Gráfica comparativa del t_{exe} del S-JPDAF en función de n , en los mismos experimentos de la Figura 7.12. Esta figura es equivalente a la Figura 6.6 del capítulo 6, obtenida para el XPFCP.	315
Figura 7.14. Tasas de error generadas por el JPDAF (en verde) y el XPFCP (en azul) en función del tipo de error, en los tests complejo (tonos claros), completo corto (tonos oscuros) y de ruido (para la tasa de ruido no filtrado).....	316

Figura 7.15. Gráficas comparativas del valor de \hat{n}_{eff} generado en el experimento complejo por el S-JPDAF y el XPFCP. Las líneas gruesas muestran el valor medio de las de trazo fino. Se incluye una tabla con el valor medio de \hat{n}_{eff} en ese test (a) y en el completo corto (b). 317

Figura 7.16. Gráficas comparativas del t_{exe} del S-JPDAF (en verde oscuro), del XPFCP (en azul) y del JPDAF (en verde claro) en el experimento complejo. 318

Figura 7.17. Gráficas comparativas del t_{exe} del S-JPDAF (en verde oscuro), del JPDAF (en verde claro) y del XPFCP (en azul) en el experimento completo largo..... 319

Figura 7.18. Proyección en la imagen de entrada de los resultados generados en el test complejo, por el JPDAF (fila superior de imágenes), el S-JPDAF (fila central) y el XPFCP (fila inferior). Se incluye el valor de la posición media estimada por cada seguidor para cada objeto detectado en las imágenes, y otros parámetros del proceso..... 321

Figura I.1. Representación funcional del GPF para la estimación de una variable unidimensional. En el paso de normalización se muestra, a modo de ejemplo, el valor estimado de la creencia $p(\hat{x}_t|y_{1:t})$ que arrojaría un GPF (en línea continua) y un GSPF (en línea discontinua). 330

Figura I.2. Representación funcional de la obtención de la función continua de regularización del set de partículas $g(S)$ característica del RPF, para un problema de estimación unidimensional. Extraída de [Doucet01]. 332

Figura I.3. Representación funcional de los algoritmos “Pre-RPF” y “Post-RPF” para la estimación de una variable unidimensional. 333

Figura II.1. Métodos de clasificación. Extraído de [Kainulainen02]. 341

Figura II.2. Ejemplo de clasificación divisiva de un conjunto de datos definidos en un espacio bidimensional, extraído de [Jain99]. 342

Figura II.3. Ejemplo de clasificación jerárquica extraído de [Jain99]..... 343

Figura II.4. Ejemplo de clasificación borrosa de un conjunto de datos definidos en un espacio bidimensional..... 343

Figura II.5. Representación del entorno de distancia constante al centroide (punto negro \bar{g}) en un espacio bidimensional ($\lambda_T = 2$), definido por la ecuación <II.2> en función de δ 345

ÍNDICE DE TABLAS

Tabla 1.1. Nivel de desarrollo, en lo que a su inserción en el mercado se refiere, de las distintos tipos de robots de servicio, en función de su área de aplicación. Extraído de [Schraft94] 3

Tabla 6.1. Reparto porcentual del número de partículas del set de salida del XPFCP S_t asociado a cada una de las cuatro hipótesis de seguimiento del experimento de la Figura 6.13. Se indica el número porcentual de partículas de los objetos numerados según aparecen ubicados de izquierda a derecha en las imágenes derechas de la Figura 6.13..... 230

Tabla 6.2. Reparto porcentual del número de partículas del set de salida del XPFCP S_t asociado a cada una de las cuatro hipótesis de seguimiento del experimento de la Figura 6.18. Se indica el número porcentual de partículas de los objetos numerados según aparecen ubicados de izquierda a derecha en las imágenes derechas de la Figura 6.13..... 235

Tabla 6.3. Valor medio de \hat{n}_{eff} generado por el XPFCP en función del algoritmo de resampling usado en los experimentos complejo y completo-corto. 241

Tabla 6.4. Valor medio e incremento absoluto y relativo del t_{exe} del XPFCP al ejecutar el experimento complejo en función del algoritmo de clasificación de medidas usado. 243

Tabla 6.5. Tasa de rechazo al ruido del XPFCP, en el experimento de ruido, en función de la habilitación del proceso de validación del clasificador de medidas incluido en el filtro. 246

Tabla 6.6. Valor medio del t_{exe} del XPFCP y del XPF al ejecutarlos en los experimentos complejo y completo-corto.....	252
Tabla 6.7. Valor medio e incremento absoluto y relativo del t_{exe} del XPFCP al ser ejecutado en el experimento complejo, en función del algoritmo de clasificación de partículas $S_t \Rightarrow G_{1:k,t out}$ usado.....	263
Tabla 6.8. Velocidad media de ejecución del XPFCP, en función del algoritmo de clasificación de partículas usado, en distintos experimentos (complejo (a), completo-corto (b) y (c) completo-largo).....	263
Tabla 6.9. Tasa de rechazo al ruido del XPFCP con clasificador de partículas, en el experimento de ruido, en función de la habilitación del proceso de validación incluido.....	266
Tabla 7.1. Valor medio absoluto y relativo del tiempo de cómputo de los procesos de asociación y estimación implicados en el JPDAF y el XPFCP al ser ejecutados en el experimento complejo.	303
Tabla 7.2. Velocidad media de ejecución del XPFCP y del JPDAF en los experimentos complejo (a), completo corto (b) y completo largo (c).....	304
Tabla 7.3. Valor medio absoluto y relativo del tiempo de cómputo de los procesos de asociación y estimación implicados en el S-JPDAF, el JPDAF y el XPFCP al ser ejecutados en el experimento complejo.....	318
Tabla 7.4. Velocidad media de ejecución del S-JPDAF, del JPDAF y del XPFCP en los experimentos complejo (a), completo corto (b) y completo largo (c)	319
Tabla III.1. Resumen de características de los experimentos usados	350

GLOSARIO

Convenciones sobre la notación de símbolos

Los acrónimos se indican en letra mayúscula

Las matrices se representan en cursiva y mayúsculas

Las variables escalares o vectoriales se representan en cursiva y minúsculas

Los vectores llevan una flecha sobre el nombre de la variable

Acrónimos

<i>APF</i>	<i>Auxiliary Particle Filter</i>	<i>Filtro de Partículas Auxiliar</i>
<i>ASIR</i>	<i>Auxiliary Sequential Important Resampling</i>	<i>Algoritmo Auxiliar de Muestreo Secuencial Ponderado con Selección</i>
<i>BPF</i>	<i>Boosted Particle Filter</i>	<i>Filtro de Partículas “Boosted”</i>
<i>BSAS</i>	<i>Basic Sequential Algorithm Scheme</i>	<i>Algoritmo de Esquema Secuencial Básico</i>
<i>CA</i>	<i>Constant Acceleration model</i>	<i>Modelo de Aceleración Constante</i>
<i>CV</i>	<i>Constant Velocity model</i>	<i>Modelo de Velocidad Constante</i>
<i>DBN</i>	<i>Dynamic Bayesian Network</i>	<i>Red Bayesiana Dinámica</i>
<i>EKF</i>	<i>Extended Kalman Filter</i>	<i>Filtro de Kalman Extendido</i>
<i>EKPF</i>	<i>Extended Kalman Particle Filter</i>	<i>Filtro de Partículas con Filtro de Kalman Extendido</i>
<i>FCM</i>	<i>Fuzzy C-Means</i>	<i>K-Medias Borroso</i>
<i>FPS</i>	<i>Frames Per Second</i>	<i>Imágenes Por Segundo</i>
<i>GPF</i>	<i>Gaussian Particle Filter</i>	<i>Filtro de Partículas Gaussiano</i>

<i>GSPF</i>	<i>Gaussian Sum Particle Filter</i>	<i>Filtro de Partículas Suma de Gaussianas</i>
<i>HMI</i>	<i>Human-Machine Interface</i>	<i>Interfaz Hombre-Máquina</i>
<i>JPDA</i>	<i>Joint Probabilistic Data Association</i>	<i>Asociación Probabilística Conjunta de Datos</i>
<i>JPDAF</i>	<i>Joint Probabilistic Data Association Filter</i>	<i>Filtro de Asociación Probabilística Conjunta de Datos</i>
<i>KF</i>	<i>Kalman Filter</i>	<i>Filtro de Kalman</i>
<i>MAP</i>	<i>Maximum a Posteriori</i>	<i>Máximo a Posteriori</i>
<i>MBSAS</i>	<i>Modified BSAS</i>	<i>BSAS Modificado</i>
<i>MCL</i>	<i>Monte-Carlo Localization</i>	<i>Localización de Monte-Carlo</i>
<i>MCMC</i>	<i>Markov Chain Monte-Carlo Step</i>	<i>Cadena de Markov con Paso de Monte-Carlo</i>
<i>MC-JPDAF</i>	<i>Monte-Carlo Joint Probabilistic Data Association Filter</i>	<i>Filtro de Monte-Carlo con Asociación Probabilística Conjunta de Datos</i>
<i>MHT</i>	<i>Multi-Hypothesis Tracking</i>	<i>Seguimiento con Hipótesis Múltiples</i>
<i>ML</i>	<i>Markov Localization</i>	<i>Localización de Markov</i>
<i>MOT</i>	<i>Moving Object Tracking</i>	<i>Seguimiento de Objetos en Movimiento</i>
<i>MP</i>	<i>Markov Process</i>	<i>Proceso de Markov</i>
<i>MPF</i>	<i>Mixture Particle Filter</i>	<i>Filtro de Partículas de Mezcla de Componentes</i>
<i>MRF</i>	<i>Markov Random Field</i>	<i>Campo Aleatorio de Markov</i>
<i>MTT</i>	<i>Multi-Target Tracking</i>	<i>Seguimiento de Múltiples Objetos</i>
<i>NN</i>	<i>Nearest Neighbour</i>	<i>Vecino Más Próximo</i>
<i>PCA</i>	<i>Principal Components Analysis</i>	<i>Análisis de Componentes Principales</i>
<i>PDA</i>	<i>Probabilistic Data Association</i>	<i>Asociación Probabilística de Datos</i>
<i>PDAF</i>	<i>Probabilistic Data Association Filter</i>	<i>Filtro de Asociación Probabilística de Datos</i>
<i>PDF</i>	<i>Probability Density Function</i>	<i>Función Densidad de Probabilidad</i>
<i>PF</i>	<i>Particle Filter</i>	<i>Filtro de Partículas</i>
<i>PMHT</i>	<i>Probabilistic Multi-Hypothesis Tracking</i>	<i>Seguimiento con Hipótesis Múltiples Probabilístico</i>
<i>POMDP</i>	<i>Partially Observable Markov Decision Process</i>	<i>Proceso de Decisión de Markov Parcialmente Observable</i>
<i>RBPF</i>	<i>Rao-Blackwellized Particle Filter</i>	<i>Filtro de Partículas “Rao-Blackwellized”</i>
<i>RPF</i>	<i>Regularized Particle Filter</i>	<i>Filtro de Partículas Regularizado</i>
<i>SIS</i>	<i>Sequential Important Sampling</i>	<i>Muestreo Secuencial Ponderado</i>
<i>SIR</i>	<i>Sequential Important Resampling</i>	<i>Muestreo Secuencial Ponderado con Selección</i>
<i>S-JPDAF</i>	<i>Sampled Joint Probabilistic Data Association Filter</i>	<i>Filtro Muestreado de Asociación Probabilística Conjunta de Datos</i>
<i>SIR-MCJPDA</i>	<i>Sequential Importance Sampling Monte-Carlo Joint Probabilistic Data Association</i>	<i>Muestreo Secuencial Ponderado de Monte-Carlo con Asociación Probabilística Conjunta de Datos</i>
<i>SMC</i>	<i>Sequential Monte-Carlo estimation</i>	<i>Estimación Secuencial de Monte-Carlo</i>
<i>SLAM</i>	<i>Simultaneous Localization and Mapping</i>	<i>Localización y Mapeo Simultáneo</i>
<i>SUT</i>	<i>Scaled Unscented Transformation</i>	<i>Transformación Escalada “Unscented”</i>
<i>TTSAS</i>	<i>Two Thresholds Sequential Algorithm Scheme</i>	<i>Algoritmo de Esquema Secuencial con Dos Límites</i>
<i>UKF</i>	<i>Unscented Kalman Filter</i>	<i>Filtro de Kalman “Unscented”</i>
<i>UPF</i>	<i>Unscented Particle Filter</i>	<i>Filtro de Partículas “Unscented”</i>
<i>XPF</i>	<i>Extended Particle Filter</i>	<i>Filtro de Partículas Extendido</i>
<i>XPFCP</i>	<i>Extended Particle Filter with Clustering Process</i>	<i>Filtro de Partículas Extendido con Proceso de Segmentación</i>

Símbolos

XYZ	Sistema de coordenadas tridimensional de componentes x , y , z que define el entorno de seguimiento.
x	Coordenada de lateralidad en sistema tridimensional que define el entorno
y	Coordenada de altura en sistema tridimensional que define el entorno
z	Coordenada de profundidad en sistema tridimensional que define el entorno
v_x	Componente de velocidad en dirección del eje x en el sistema de coordenadas XYZ
v_y	Componente de velocidad en dirección del eje y en el sistema de coordenadas XYZ
v_z	Componente de velocidad en dirección del eje z en el sistema de coordenadas XYZ
UV	Sistema de coordenadas bidimensional de componentes u , v que define el plano imagen de la cámara.
u	Coordenada de abscisas en sistema bidimensional que define el plano imagen
v	Coordenada de ordenadas en sistema bidimensional que define el plano imagen
I	Matriz identidad
\vec{x}	Vector de estado de un sistema descrito en variables de estado ¹
χ	Vector de estado extendido, que incluye el del conjunto de k objetos implicados en una aplicación de seguimiento de múltiples objetos $\chi_t = \{\vec{x}_{1,t}, \vec{x}_{2,t}, \dots, \vec{x}_{k,t}\}$
A	Matriz de transición de un sistema lineal descrito en variables de estado
\vec{u}	Vector de entrada de un sistema descrito en variables de estado
B	Matriz de transición de entradas de un sistema expresado en variables de estado
m	Número total de medidas
\vec{y}	Vector de salida (o de medida) de un sistema descrito en variables de estado
Y	Matriz de medidas generado por el sistema de seguimiento. $Y = \{\vec{y}_i / i = 1 : m\}$
C	Matriz de transición de un sistema lineal descrito en variables de estado
D	Matriz de realimentación de un sistema lineal descrito en variables de estado
\vec{v}	Vector de ruido asociado al de estado en un sistema descrito en variables de estado
V	Matriz de covarianza del ruido de estado
\vec{o}	Vector de ruido asociado al de medidas en un sistema descrito en variables de estado
O	Matriz de covarianza del ruido de medida
$f(\vec{x}_t, \vec{u}_t, \vec{o}_t)$	Función no lineal que caracteriza a la ecuación de transición del modelo de un sistema expresado en variables de estado
$h(\vec{x}_t, \vec{u}_t, \vec{r}_t)$	Función no lineal que caracteriza a la ecuación de salida del modelo de un sistema expresado en variables de estado
λ	Cada una de las dimensiones del espacio de características de la clasificación
k	Número de clases. Se añade el subíndice “in” para indicar clases de medidas y “out” para indicar clases de partículas, o número de objetos en la aplicación de seguimiento
G	Matriz/estructura contenedora de la información de salida del proceso de clasificación para cada clase. $G_j / j = 1 : k$

¹ La flecha indicativa de vector diferencia a los vectores \vec{u} , \vec{v} , \vec{x} e \vec{y} de los valores escalares de coordenadas u , v , x e y de los espacios UV y XYZ , el número total de miembros l_j de cada clase j de la variable que caracteriza a cada uno de sus i miembros $\vec{l}_{i,j} / i = 1 : l_j$, y su centroide \vec{g}_j con la función genérica $g()$.

\bar{g}_j	Centroide de la clase. $\bar{g}_j / j = 1:k$
L_j	Conjunto de miembros de la clase. $L_j = \{\bar{l}_{i,j} / i = 1:l_j\}$, donde $\bar{l}_{i,j}$ representa a cada miembro de la clase y l_j al número total de miembros de la misma
τ_j	Identificador de la clase. $\tau_j / j = 1:k$
β_j	Variable de validación de la clase. $\beta_j / j = 1:k$
α	Factor de ponderación de verosimilitud de las clases: α_{new} para crear nuevas clases con el clasificador “subtractive” α_{valid} factor de olvido para validar las clases
π	Factor de histéresis de los límites de validación de clases: π_d para el límite de distancia d_{valid} π_p para el límite de verosimilitud p_{valid}
n	Número total de partículas
n_{eff}	Número total de partículas efectivo según [Liu98]
\bar{s}	Valor de una partícula.
S	Matriz de partículas. $S = \{\bar{s}_i / i = 1:n\}$
\bar{w}	Vector de pesos del conjunto de partículas. $\bar{w} = \{w_i / i = 1:n\}$
$w()$	Función de ponderación del conjunto de partículas
n_m	Número total de partículas introducidas en etapa de re-inicialización del XPFCP
γ	Relación entre n_m y n
K	Matriz de estimación del KF
P	Matriz de covarianza del error de estimación en el KF
\bar{r}	Vector de residuos usado en la etapa de corrección de un seguidor basado en el KF
θ	Hipótesis de asociación disjunta de una medida $i = 1:m$ con un objeto $j = 1:k$
Λ	Verosimilitud disjunta de la hipótesis de asociación correspondiente θ
Ψ	Suma de verosimilitudes disjuntas Λ de un objeto o una medida en un evento de asociación Φ
Φ	Evento de asociación conjunta. $\Phi \subseteq \{\theta_{i,j} / i = 1:m, j = 1:k\}$
p	Valor de probabilidad o verosimilitud
$p(a b)$	Probabilidad de a condicionada a b
$p(a b,c)$	Probabilidad de a condicionada a b y c
$p(\bar{x}_t \bar{x}_{t-1})$	PDF que caracteriza al modelo de actuación o de estado en un sistema estocástico
$p(\bar{y}_t \bar{x}_t)$	Verosimilitud del vector de medidas. Caracteriza al modelo de observación o de medida en un sistema estocástico
$p(\bar{x}_t \bar{y}_{1:t-1})$	“Prior distribution” o probabilidad a priori de la estimación del vector de estado
$p(\bar{x}_t \bar{y}_{1:t})$	“Posterior distribution” o probabilidad a posteriori de la estimación del vector de estado, también llamada creencia
$E()$	Función estadística de la esperanza o el valor medio
$O()$	Complejidad computacional

μ, \mathbf{N}	<i>Media de una variable aleatoria escalar y vectorial, respectivamente</i>
σ, Ω	<i>Desviación típica de una variable aleatoria escalar y vectorial, respectivamente</i>
$norm(\mu, \sigma)$	<i>PDF normal de una variable y vector de variables aleatorias, de media μ y \mathbf{N}, y</i>
$norm(\mathbf{N}, \Omega)$	<i>desviación típica σ y Ω, respectivamente</i>
$g()$	<i>Función genérica</i>
max	<i>Función máximo</i>
min	<i>Función mínimo</i>
η	<i>Factor de normalización</i>
d	<i>Distancia genérica</i>
i, j, a	<i>Variables índice genéricas</i>
t	<i>Instante de tiempo</i>
t_s	<i>Periodo de muestreo</i>
t_{exe}	<i>Tiempo de ejecución</i>

Superíndices

T	<i>Transpuesta de una matriz</i>
\rightarrow	<i>Vector</i>
$-$	<i>Valor medio o promediado. También aparece expresado a través de la función estadística de la esperanza. $E(a) = \bar{a}$</i>
(i)	<i>Referente a una muestra de la señal</i>
\sim	<i>Variable normalizada</i>
$'$, $\hat{\sim}$	<i>Variable secundaria o aproximada</i>

Subíndices

T	<i>Valor total</i>
max	<i>Valor máximo</i>
min	<i>Valor mínimo</i>
opt	<i>Valor óptimo</i>
eff	<i>Valor eficaz</i>
exe	<i>Valor de ejecución</i>
th	<i>Valor límite</i>
valid	<i>Valor de validación</i>
cell	<i>Referente a una celda de un espacio enrejillado</i>
t	<i>Referente al instante temporal</i>
iter	<i>Referente a la iteración del algoritmo</i>
s	<i>Referente al periodo de muestreo</i>
m	<i>Referente a las medidas obtenidas con el modelo de observación</i>
g	<i>Referente al centroide del proceso de clasificación</i>

CAPÍTULO 1

INTRODUCCIÓN A LA TESIS

La innovación tecnológica es el factor más apremiante en el mundo de la investigación actual. En este ámbito, la electrónica y la inteligencia artificial, son algunas de las ciencias de más rápida evolución, pues están directamente implicadas en áreas de mucho interés para el ser humano, como son la medicina, la gestión de recursos naturales, la comunicación y la gestión de la información.

La tesis que aquí se introduce está enmarcada en el área de la robótica, cuya evolución está directamente relacionada sobre todo con la de las dos ciencias comentadas anteriormente. El párrafo inicial contextualiza y justifica el desarrollo de esta tesis.

La robótica, cuyos primeros resultados, llamados “*robots*”, sirvieron para rentabilizar los trabajos que implicaban repetibilidad o especial fuerza o precisión para el ser humano (lo que se conocía como “*robótica industrial*” en la década de los 60 del siglo pasado), se ha orientado actualmente hacia aplicaciones de servicio y gestión. Estas máquinas han pasado de ser sistemas reactivos programables a poseer autonomía e inteligencia, entendida ésta como capacidad de comprender y tomar decisiones.

Los trabajos desarrollados en la tesis se incluyen dentro de esta última generación de la robótica que se ha dado en llamar “*robótica de servicio*”.

Dentro de esta área, una serie de innovaciones todavía en fase de desarrollo permitirán a los robots de servicio desenvolverse por entornos desconocidos cada vez más complejos, gracias a su capacidad de recuperar e interpretar gran cantidad de información del entorno. Este nuevo

tipo de robots se denomina “*robots personales*” y las aportaciones presentadas en esta tesis están más concretamente orientadas a proponer soluciones para alcanzar las capacidades de este nuevo tipo de sistemas.

Tal y como se muestra en los apartados siguientes, el mayor escollo con el que se encuentra la evolución de esta última familia de robots es el conocimiento del entorno. Esta es la justificación del desarrollo de la tesis, que pretende aportar una solución al problema de la *localización y estimación del movimiento de los múltiples objetos* estáticos o que se mueven alrededor del robot en un entorno que, en el encuadre de la robótica personal, es desconocido, complejo, muy dinámico y generalmente interior.

En este capítulo de introducción a la tesis se pretende exponer de forma somera algunas características del área de investigación de la robótica de servicio y personal, con objeto de enmarcar la tesis, analizar las carencias de trabajos anteriores realizados en esta misma línea y tenerlas en cuenta en el desarrollo de los distintos algoritmos y sistemas aportados en esta tesis.

A partir de este estudio, se presentan las principales líneas de desarrollo abordadas en la tesis, justificando su interés dentro del contexto científico establecido. Este análisis permite establecer de forma esquemática los objetivos más importantes que se persiguen y alcanzan en la tesis. Finalmente se describe la estructura de la tesis con objeto de proporcionar al lector una imagen general del contenido de esta memoria.

1.1. Marco de desarrollo de la tesis

El concepto de un robot, es el de:

“una máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a las personas”¹.

El concepto es válido desde los primeros robots industriales hasta los últimos modelos que ahora se conocen, pero la constitución y funcionalidad del sistema al que este término se refiere ha cambiado enormemente de unos a otros ejemplos. La aparición de nuevas aplicaciones para la robótica llega a mediados de la década de los 80 del pasado siglo, momento en el que su objetivo de aplicabilidad deja de ser el incremento de la productividad en las fábricas y pasa a ser el servicio a la población.

1.1.1. De la robótica industrial a la de servicio y personal

La razón de evolución de la robótica se ha de buscar en el hecho de que, tras la expansión de la robótica industrial, el uso de robots de este tipo no vuelve a experimentar ningún crecimiento, pues desde ese momento cubre todas las disciplinas implicadas en el proceso de fabricación en las que la máquina es rentable. Se busca entonces una nueva aplicación para la robótica, que se encuentra en el área de máxima evolución del momento: el área de servicios. Joseph F. Engelberger, considerado uno de los padres de la robótica industrial, ya predijo este

¹ Definición del diccionario de la lengua de la Real Academia Española. 2008.

movimiento, en contra de lo que en aquel momento defendían los investigadores clásicos de robótica industrial, y así lo expuso en su obra “*Robotics in Service*” en 1989.

En [Schraft94] se hace un análisis sobre el nivel de desarrollo y el momento de incorporación al mercado de los distintos tipos de robots de servicio, según el área de aplicación o funcionalidad. La Tabla 1.1 muestra un resumen de la información que el autor recopiló en el año de publicación del artículo (1994).

Tabla 1.1. Nivel de desarrollo, en lo que a su inserción en el mercado se refiere, de las distintos tipos de robots de servicio, en función de su área de aplicación. Extraído de [Schraft94]

	En funcionamiento	En prueba (corto-medio plazo)	En concepto (largo plazo)
Entretenimiento	Máquina recoge-pelotas		
Agricultura y Construcción	Segadora automática	Mantenimiento y reparación	
Medicina	Sillas de ruedas semi-autónomas	Asistencia en operaciones	
Trabajos Peligrosos y Seguridad	Vehículos de vigilancia		Rescate autónomo
Transporte	Transporta-equipajes automático	Cinturón de seguridad	Vehículo autónomo
Limpieza y Asistencia	Aspirador autónomo	Chico de los recados autónomo	ROBOT PERSONAL

En esta Tabla 1.1 se puede ver que la evolución real que ha sufrido la robótica de servicio, a lo largo de los 10 últimos años, se corresponde con lo que el autor pronosticaba, pues la situación actual en este campo se corresponde con la predicción de corto plazo de la tabla. Además, resulta interesante observar que, según este pronóstico, los sistemas que todavía no se encuentran en el mercado se relacionan principalmente por lo que se ha dado en llamar “*robótica personal*”.

Si bien es difícil encontrar una definición de robótica personal que abarque todas las disciplinas que ésta implica, todas ellas² coinciden en matizar que el término se refiere a sistemas que suplanten o ayuden en su labor a las personas en aplicaciones de servicio o sociales en lugar de en trabajos industriales.

El propio autor caracteriza a los robots personales, entre otros factores, por ser capaces de *comprender el entorno*. En esta línea y en el mismo artículo el autor muestra un gráfico comparativo entre el robot industrial, el de servicio y el personal, en función de dos parámetros: su nivel de autonomía y la complejidad del entorno por el que se puede desenvolver. La Figura 1.1 reproduce este gráfico, y a partir de él se define la robótica personal como una evolución de la de servicio, en la que la autonomía del robot alcanza su máximo exponente en entornos cada vez más complejos, gracias a una mayor capacidad de interacción del robot con el entorno y a una mejor interpretación de la información contenida en éste por parte de la máquina.

² Ver definición de la IPA (“*Institut für Produktionstechnik und Automatisierung*” o “*Instituto de Producción y Automatización*”) <http://www.ipa.fhg.de/>, o de la IFR (“*International Federation of Robotics*”, “*Federación Internacional de Robótica*”) <http://www.ifr.org>.

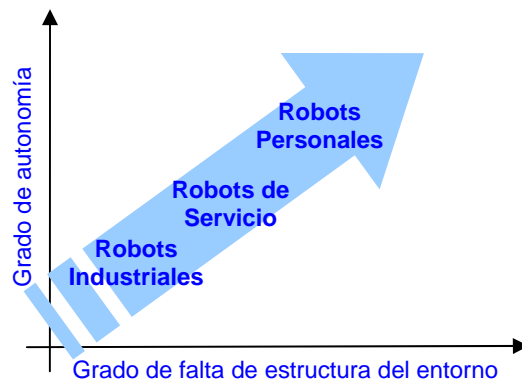


Figura 1.1. Comparativa de la robótica industrial, de servicio y personal en función de la autonomía del robot y la complejidad del entorno en el que se puede desenvolver. Reproducción de [Schraft94]

Si bien el autor incluye los robots personales únicamente en el área de asistencia en su especulación particular sobre la evolución de la robótica de servicio, actualmente es evidente que este tipo de sistemas encuentran aplicación en muchos otros sectores como: tele-vigilancia y seguridad ([Nicoud96], [Wang96] u [Ocaña05]), trabajos peligrosos o duros para las personas ([Hirose00], [Cepolina03] o [TRILOBITE]), agricultura y construcción ([Peñin98] o [Stenz02]), transporte ([PATH] o [Sotelo01]), en el sector emergente de la tele-medicina ([DA VINCI]), el de entretenimiento y acompañamiento ([MINERVA], [Fujita98] o [Ishida03]), y, por supuesto, el asistencial ([Maxwel99], [Prassler01], [García01], [NURSEBOT], [CARE-O-BOT] o [López04]).

Con todo ello se comprueba que para fomentar la evolución de la robótica en la actualidad es necesario realizar un especial esfuerzo de investigación y desarrollo en el área de la robótica personal, lo que implica dotar a los robots de la capacidad de desenvolverse en entornos complejos y sin estructura. Se trata de un objetivo no resuelto de forma global por el momento y que, tal y como ya se ha mostrado, abre un nuevo camino de aplicación para los robots.

Las aportaciones presentadas en esta tesis están orientadas a resolver al menos parcialmente este paradigma que plantea la evolución de la robótica personal, pues su objetivo principal es el de presentar un método de detección y seguimiento de objetos en entornos desconocidos, no estructurados, dinámicos y muy poblados.

1.1.2. Necesidades de la robótica personal

La necesidad de que el robot sea capaz de adaptarse a entornos cambiantes y complejos, que surge como característica más relevante de la robótica personal, plantea establecer una serie de restricciones de necesario cumplimiento en este tipo de máquinas:

- *Movilidad*: El robot pasa de ser un elemento estático o de recorrer rutas predefinidas, en las aplicaciones industriales, a requerir capacidad de movimiento autónomo por entornos desconocidos. Este cambio es quizás el más importante a nivel técnico en la evolución de estas máquinas pues afecta a distintas disciplinas incluidas todas ellas en lo que se viene a llamar “navegación autónoma” del robot: *gestión de comportamientos, mapeo del entorno*

de movimiento, localización en el mismo, planificación de rutas y generación y control de trayectorias de movimiento. La tarea de localización y detección de objetos en el entorno no ha sido abordado de forma independiente hasta hace pocos años, pues que se incluía como una subtarea reactiva más, generalmente dentro de la de control de trayectorias del movimiento del robot como sistema de evasión de obstáculos en la trayectoria planificada. Las necesidades impuestas por las aplicaciones de la robótica de servicio y personal provocan la búsqueda de soluciones deliberativas para esta tarea, que realmente está relacionada con todas las tareas implicadas en la navegación autónoma del robot (localización, mapeo y planificación de rutas además de la de control de movimiento).

- *Extracción de información del entorno:* Al poder ser desconocido el entorno de movimiento, el robot personal ha de incorporar la capacidad de obtener información del mismo, más completa, en general, que la que necesaria en una aplicación industrial. Con los datos obtenidos del entorno se procede a construir una base de conocimiento del mismo de forma local o parcial (implicando de nuevo tareas de detección y seguimiento de objetos) o total (mapeado del entorno), que puede usarse en tareas como la de navegación, ya expuesta, o en otras de más reciente creación, al ser propias de la robótica personal, como la interacción con el usuario (base de los “*Sistemas de Interacción Hombre-Máquina*” o “*Human-Machine Interfaces*”, HMIs) o el *aprendizaje de comportamientos* (propio de la “*Robótica Basada en Comportamientos*” o “*Behaviour-Based Robotics*”).
- *Seguridad:* Uno de los mayores retos que tiene que superar la robótica en su integración en el área personal es el de dotar a las máquinas de sistemas de seguridad suficientes como para hacer que los humanos se sientan cómodos en su interacción con los robots. La tarea de detección de obstáculos que dificulten la maniobra del robot es de nuevo una de las más implicadas en este objetivo, del mismo modo que la de los HMIs ya mencionados.

El papel del conocimiento del entorno es común a todas las especificaciones expuestas, y, en opinión de la autora de esta tesis, el presente y futuro de la robótica como área importante de investigación aplicada está en que el robot sea capaz de obtener e interpretar, en tiempo real (Es decir, a la velocidad de entrada de la información en el sistema), tanta información del entorno como la que recoge el ser humano, para posteriormente tomar decisiones a partir de ésta.

Partiendo de este hecho, la evolución tecnológica, en lo que se refiere a incremento de la velocidad en el manejo de grandes cantidades de información, ha supuesto el punto de partida necesario para la orientación de la investigación robótica a la aplicación personal. Esta es quizás una de las razones más importantes que justifican la corta vida de la robótica personal aplicada.

Debido a ello, las soluciones propuestas por la comunidad científica para el conocimiento y reconstrucción del entorno del robot son quizás las que presentan todavía una solución global menos desarrollada, por lo que las aportaciones de la comunidad científica en este apartado son especialmente valiosas. En este marco, la tarea de SLAM (“*Simultaneous Localization and Mapping*”, “*Localización y Mapeo Simultáneo*” ([Leonard91], [Castellanos99] o [Durrant-Whyte06]) es actualmente objeto de frecuentes trabajos pues resuelve el problema de que el robot reconstruya un entorno desconocido mientras se mueve y posiciona en él. Por la misma

causa, los trabajos relacionados con la tarea de la detección y el seguimiento de la posición de los elementos dinámicos que aparecen en el entorno del robot, objetivo principal de esta tesis, centran también los esfuerzos de los investigadores actuales y se convierte en una tarea imprescindible en la implementación de un robot personal autónomo, y además, en una de las más presentes en los desarrollos científicos actuales, tal y como se muestra en el capítulo 2.

1.1.3. Modelando la incertidumbre

Uno de los escollos más importantes con los que la comunidad científica se ha encontrado al extraer datos del entorno aparece a la hora de intentar adaptar la información sensada a un modelo que en los trabajos de robótica industrial se plantea de forma determinística y estática.

En este punto tiene un papel muy importante el uso de modelos y algoritmos nuevos que incorporen incertidumbre a la observación y la acción del robot ante el entorno, frente a los modelos determinísticos propios de la segunda generación de robots (lo que se conoce como “*arquitecturas deliberativas*”, [Arkin98]). Así surge el uso de los “*métodos u algoritmos probabilísticos*” en la programación de las distintas funcionalidades de los robots, que se ha revelado como uno de los avances más interesantes en la aportación de una mayor capacidad de entendimiento del entorno a estos sistemas.

Este tipo de algoritmos se basan en la teoría de probabilidad y más concretamente en el “*filtro de Bayes*” ([Thrun00]), cuya formalización se remonta a la década de los 60. Sin embargo su incorporación al área de la robótica se produjo alrededor de los años 90, y es justamente en el área de la robótica personal, donde se han revelado con más éxito, pues su uso arrancó el cambio en las aplicaciones de robótica de servicio hacia el área personal.

Merece la pena destacar que los modelos probabilísticos emulan el comportamiento del humano, un comportamiento estocástico y ruidoso que se adapta a las condiciones que le rodean. En esta simulación radica el éxito de los trabajos de robótica personal que incluyen métodos probabilísticos para programar las distintas funcionalidades del robot.

Los métodos probabilísticos se han aplicado con éxito a diferentes tareas del comportamiento de un robot en el diseño de aplicaciones personales: localización, mapeo y SLAM, planificación de rutas, gestión de comportamientos, interacción con el usuario (en interfaces vocales, sobre todo), etc. En lo que se refiere a la aplicación de interés en la tesis, el uso de algoritmos probabilísticos es un hecho muy reciente que pasa por la exportación de las soluciones ya planteadas de otras áreas (sobre todo las de localización y mapeo) a la de detección y seguimiento de objetos ([Thrun00] o [Fox03]).

La versión de algoritmo probabilístico depende, evidentemente, de la funcionalidad robótica a resolver y de si las variables implicadas en la tarea son continuas (sobre todo en funciones de posicionamiento o movimiento) o discretas (más relacionadas con tomas de decisión) hablando así de un enfoque métrico o topológico, respectivamente, del algoritmo.

De este modo las distintas particularizaciones dan lugar a algoritmos probabilísticos muy diferentes ([Arulampalam02]), cuya aplicación en distintas áreas de la robótica está siendo en general rápidamente extendida entre la comunidad científica:

- Desde la versión óptima del filtro de Bayes, que se obtiene de aplicar una serie de restricciones al modelo estocástico implicado. El resultado es el conocido filtro de Kalman (“*Kalman Filter*”, KF, [Kalman60]), que por su sencilla implementación se utiliza en muy diferentes áreas de tecnología, informática, automatización, matemática o estadística y, por supuesto, robótica.
- Hasta versiones aplicables a problemas que no cumplen tales restricciones, por lo que se utilizan en tareas más complejas de inteligencia artificial, o robótica personal. Ejemplos de estos últimos son el filtro de partículas (“*Particle Filter*”, PF, [Gordon93]), los POMDPs (“*Partially Observable Markov Decision Process*”, o “*Proceso de Decisión de Markov Parcialmente Observable*”) y las DBN (“*Dynamic Bayesian Network*”, o “*Red Bayesiana Dinámica*”).

En cualquiera de los casos la base del algoritmo es, tal y como ya se ha comentado, el filtro bayesiano, caracterizado por sus dos etapas de ejecución (“*predicción*” y “*corrección*”) y por implicar una representación continua o discreta de cualquier variable involucrada en el proceso como una “*función densidad de probabilidad*”, PDF.

La forma de las PDFs involucradas en los filtros bayesianos es lo que diferencia a sus diferentes implementaciones prácticas, desde un punto de vista funcional. La PDF en su forma más general es una función continua como la mostrada en la imagen izquierda de la Figura 1.2, $p(x)$, PDF de la variable unidimensional x . Sin embargo, el tratamiento de estas variables es matemáticamente imposible de forma práctica, por lo que se usan otros formatos para las PDFs que caracterizan a las variables implicadas, como el continuo y parametrizable propio del KF ($p(x) = \text{norm}(\mu, \sigma)$ en la imagen central en la figura), o el discreto, propia de los PFs, POMDPs y DBNs (imagen izquierda en la figura).

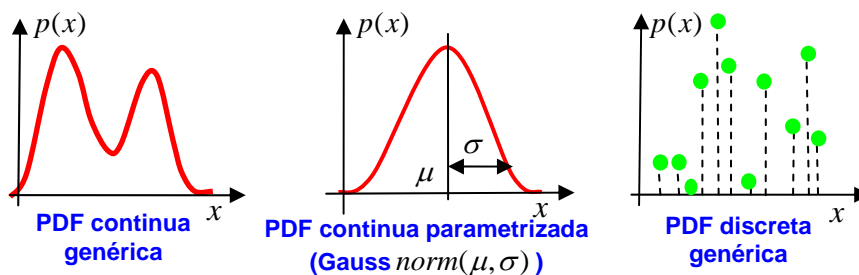


Figura 1.2. Ejemplo de las formas más habituales de la PDF $p(x)$ en los algoritmos probabilísticos: continua genérica a la izquierda; continua parametrizable ($p(x) = \text{norm}(\mu, \sigma)$), propia del KF en el centro; discreta genérica propia del PF a la derecha.

Es necesario analizar la ventaja o inconveniente del trabajo con una u otra forma, a nivel de tiempo de ejecución, exactitud y robustez del algoritmo probabilístico resultante, para decidir

cual es la alternativa más adecuada para cada aplicación. Este análisis se incluye en el capítulo 2, para la aplicación de seguimiento de múltiples objetos de interés en la tesis.

Este modelo estocástico, asociado a las variables implicadas en los algoritmos probabilísticos, permite incorporar un valor de incertidumbre o ruido al proceso en que éstos se empleen, lo cual revierte en un incremento de su robustez, flexibilidad y fiabilidad, especificaciones establecidas como imprescindibles en aplicaciones de robótica personal.

En esta tesis se propone la combinación de métodos probabilísticos y determinísticos para resolver el problema de localización y seguimiento de objetos en aplicaciones de robótica personal. Tal y como se justifica en la tesis, esta combinación mejora la robustez y fiabilidad del algoritmo de MTT (“*multiple target tracking*”, “*seguimiento de múltiples objetos*”) propuesto y minimiza su carga computacional de modo que pueda ser ejecutado en tiempo real.

En el capítulo 4 se realiza una revisión más completa sobre la teoría de estimación bayesiana desde el punto de vista del trabajo aquí expuesto. Baste en este apartado con resaltar que los métodos probabilísticos aparecen en la literatura científica bajo diversas acepciones. Así términos como: estimador, algoritmo o filtro bayesiano, probabilístico o de Monte-Carlo, se usan para referirse a la misma cuestión.

1.1.4. Robots personales en entornos densamente poblados

De todas las tareas implicadas en la funcionalidad de un robot personal, es la de localización la que presenta mayor relación con el desarrollo de la tesis. El problema aquí planteado (estimar la posición y el movimiento de múltiples objetos en entornos interiores y complejos) es, en principio, un problema de localización local, con la complejidad no siempre contemplada en los trabajos de este tipo realizados por la comunidad investigadora, de incluir un número desconocido y variable de elementos a localizar, de posición inicial desconocida.

Por otra parte, la implicación de unas condiciones complejas del entorno de movimiento en las tareas de detección, posicionamiento y seguimiento no ha sido extensamente abordada hasta los últimos diez años y son pocos los grupos de investigación que han desarrollado proyectos que resuelven con éxito este problema, y cuando lo hacen es bajo condiciones muy concretas.

En este apartado se recogen los ejemplos más importantes desarrollados por la comunidad científica que se aproximan al concepto de robot personal. El objetivo es analizar funcionalmente si incluyen el módulo de MTT propuesto en esta tesis y bajo qué condiciones se desarrollan (entorno, algoritmia, etc.), y justificar así el interés de la tesis. El análisis exhaustivo de las técnicas y algoritmos empleados en cada caso se deja para el capítulo siguiente.

Investigadores de la universidad americana Carnegie Mellon en colaboración con otros de la Universidad de Bonn, han desarrollaron en el periodo 1997 a 2002 diferentes robots personales, con distintas aplicaciones:

- En los años 1997 y 1999 los robots guías de museo Rhino ([RHINO]) y Minerva ([MINERVA]) (Figura 1.3).



Figura 1.3. Robots Minerva y Rhino actuando como guías de los museos Smithsonian's National Museum of American History en Washington (USA) y Deutsches Museum en Bonn (Alemania) respectivamente.

- Posteriormente (año 2002) un grupo de trabajo de la Universidad de Bonn en colaboración con instituciones como la Universidad de Freiburg, presenta nuevos robots guías de museo, bajo los proyectos Tourbot y WebFair ([TOURBOT] y [Trahancias05]) (Figura 1.4).



Figura 1.4. Robots del proyecto Tourbot en distintas situaciones: Lefkos en la Foundation of the Hellenic World, en Atenas (Grecia) y Albert en la feria Heinz Nixdorf Museums-Forum en Paderborn (Alemania).

- Finalmente el grupo de la universidad Carnegie Mellon diseña a continuación de Minerva (año 2001 y 2002) los robots de asistencia personal Flo ([Roy00]), Pearl ([Pineau03] [Pollack02]) desarrollados bajo el proyecto Nursebot ([NURSEBOT]) (Figura 1.5).



Figura 1.5. Robots Flo y Pearl interactuando con personas. Flo en un programa televisivo de la NBC y Pearl en un centro de personas mayores en Longwood (USA).

En todos estos proyectos se utilizan algoritmos probabilísticos para localizar al robot en el entorno por el que se desplaza. En todos los casos el problema de la detección de objetos alrededor del robot se resuelve usando sensores de ultrasonidos, láser, y en el caso de Tourbot y WebFair cierta información de visión.

En casi ninguno de los robots se incluye una tarea de seguimiento de estos objetos. La excepción se presenta en los últimos prototipos del proyecto Tourbot, en los que se incorpora la tarea de localización de las personas que se detecten alrededor del robot mediante información de un láser de infrarrojos y se sigue su movimiento gracias a un estimador bayesiano.

La otra línea de proyectos importante en la que se plantea resolver el problema de navegación autónoma en entornos densamente poblados de un robot móvil, es la desarrollada por investigadores del Instituto de Investigación FAW de Alemania y del Instituto Tecnológico de California, en el diseño de la silla de ruedas autónoma Maid ([Prassler01]). Uno de los objetivos del trabajo es proporcionar movilidad a usuarios de sillas de ruedas en entornos no estructurados y muy poblados, en los que éstos encuentran especial dificultad de maniobra.

En este trabajo tampoco se resuelve el problema de seguimiento de múltiples objetos, sino que al igual que hacen los robots Minerva y Rhino, Maid únicamente detecta con sensores de proximidad (láser infrarrojo y ultrasonidos) los objetos que se encuentran en su entorno y, utilizando un mapa ocupación de rejilla con marcas temporales, determina cuáles son estáticos y cuáles dinámicos, y su vector velocidad característico en este último caso.

En la misma línea de proyectos, se aborda más tarde ([Illmann01]) el objetivo de que la silla de ruedas siga a una persona por un entorno muy poblado, pero como se trata de seguir un único objeto el problema no es comparable con el objetivo planteado en esta tesis.

La Figura 1.6 muestra una imagen de la silla de ruedas moviéndose de forma autónoma en la estación de tren de Ulm (Alemania), donde estuvo en fase de pruebas durante varios días con distinta intensidad de tráfico de personas, con resultados exitosos. En la misma figura se observa también una imagen de Maid desarrollado tareas de seguimiento de una persona.



Figura 1.6. Silla de ruedas Maid en movimiento autónomo por la estación de ferrocarril de Ulm (Alemania).

Posteriormente, en los años 2001 a 2003, los autores de [Prassler01], en este caso trabajando en el “*Instituto de Producción y Automatización*” (“*Institut für Produktionstechnik und Automatisierung*”, IPA) en Stuttgart (Alemania), presentan varios prototipos de robots asistenciales bajo el proyecto Care-o-bot ([CARE-O-BOT]), pero ninguno de ellos resuelve el problema de seguimiento de objetos de forma específica ([Graf04]).

De la exposición anterior se extrae la conclusión de que la tarea de detección y seguimiento de múltiples objetos en entornos densamente poblados no ha sido abordada hasta los últimos

años, y cuando se ha hecho ha sido en contextos muy particulares. En cualquier caso, las soluciones aportadas han sido pocas y con escaso éxito, lo cual justifica el interés de los trabajos desarrollados en esta tesis.

Finalmente, es necesario destacar que, si bien la característica principal que hace a los métodos probabilísticos exitosos en la resolución de problemas de estimación es su capacidad de integrar en el proceso el valor de incertidumbre de las variables asociadas, la particularidad que los hace, aún si cabe, más interesantes para la resolución del problema que se plantea en la tesis es su capacidad de manejar procesos multimodales³.

Esta característica de los algoritmos probabilísticos, que se analiza más adelante, no ha sido explotada en la mayor parte de las soluciones aportadas por la comunidad científica al problema de seguimiento de múltiples objetos, tal y como se muestra en el capítulo 2 de esta tesis. Generalmente, esta tarea se resuelve utilizando un estimador para cada uno de los objetos a seguir, o algún método aproximado a la formulación de Bayes sin un planteamiento teórico formal, como ocurre en el proyecto de la silla de ruedas Maid.

El uso de un único estimador que sea capaz de manejar sistemas multimodales en el algoritmo de seguimiento de múltiples objetos tiene numerosas ventajas, tal y como se demostrará a lo largo de la tesis; y los métodos probabilísticos son, hoy por hoy, la única solución robusta para implementar esta tarea en aplicaciones de robótica personal en tiempo real, en las que el entorno del robot es muy complejo y la robustez, seguridad y tolerancia a los fallos del sistema es un factor primordial.

1.1.5. Sistemas sensoriales para la extracción de la información

En esta tesis, el concepto de *extracción de características de los objetos del entorno* se refiere a la obtención del vector de medidas de posición respecto a un sistema de coordenadas local a partir de información sensorial que se utiliza como entrada del algoritmo de seguimiento.

De apartados anteriores se concluye que la extracción y el procesado de la mayor cantidad posible de información del medio es, probablemente, el factor clave para el desarrollo de aplicaciones de robótica personal.

Debido a este hecho, y superada la limitación que suponía el manejo de grandes cantidades de datos gracias a la evolución tecnológica, el uso de sensores de visión para la captación de información en este tipo de aplicaciones, se ha extendido con rapidez y éxito durante las dos últimas décadas. Ello se debe a que la misma evolución tecnológica ha convertido a las cámaras en sensores que permiten obtener gran cantidad de información del entorno de movimiento del robot, de mucha calidad, y a muy bajo coste y alta velocidad.

Por otro lado, la fusión multisensorial se ha revelado como una mejora muy interesante para el robustecimiento de distintos procesos relacionados con la navegación de robots. En el caso de

³ Se denominan sistemas multimodales a aquellos que permiten caracterizar varios procesos con un único modelo. Los modelos bayesianos son multimodales pues permiten caracterizar distintas variables con una única función densidad de probabilidad. Al final de este capítulo se analiza en detalle el significado de este término en la tesis.

la localización, la posibilidad de procesar información procedente de distintos tipos de sensores de distancia (por señal ultrasónica, infrarroja, de visión o de radio) debe estar presente en cualquier solución si se desea dotarla de cierta fiabilidad, pues en general cada sensor tiene características que lo hacen más adecuado para resolver el problema de la localización en una situación de entorno concreta.

En esta tesis, y con objeto de realizar un análisis de aplicabilidad lo más completo posible de los algoritmos propuestos, se plantea un modelo general para el sistema de sensado que se utiliza en la extracción de información del entorno. El uso de métodos probabilísticos en el proceso diseñado en la tesis, para resolver el problema de localización y seguimiento de objetos, facilita la aplicación de este modelo genérico. Sin embargo, para poder obtener resultados reales de implementación de los algoritmos propuestos, se ha utilizado un sistema de visión para extraer la información del entorno, teniendo en cuenta lo expuesto en párrafos anteriores.

Tal y como se muestra en el capítulo siguiente, existen múltiples propuestas planteadas por la comunidad científica a lo largo de las dos últimas décadas para resolver el problema de MTT con información de visión. La mayor parte de ellas se centran en aprovechar al máximo las técnicas de procesado de la imagen para obtener un modelo específico y robusto de los objetos a seguir, basado en el color, el contorno, la forma, etc. En contrapartida, en muchos casos se simplifica el algoritmo de estimación de posición. En línea con las ideas de fusión multisensorial y tipología variada de los objetos a seguir comentadas, en esta tesis se plantea trabajar con modelos de objetos simples y generales, adaptables a casi cualquier tipo de objeto y sistema de observación involucrado en la tarea de seguimiento, si bien, como ya se ha dicho, se usan medidas de visión para poner en práctica los algoritmos diseñados.

A partir de esa idea se supone que el set de medidas de posición extraídas del entorno es un conjunto de puntos más o menos denso, desigualmente distribuido entre los objetos de distinta tipología, tamaño y dinámica, y con contenido de ruido como el que se muestra en la imagen izquierda de la Figura 1.7.

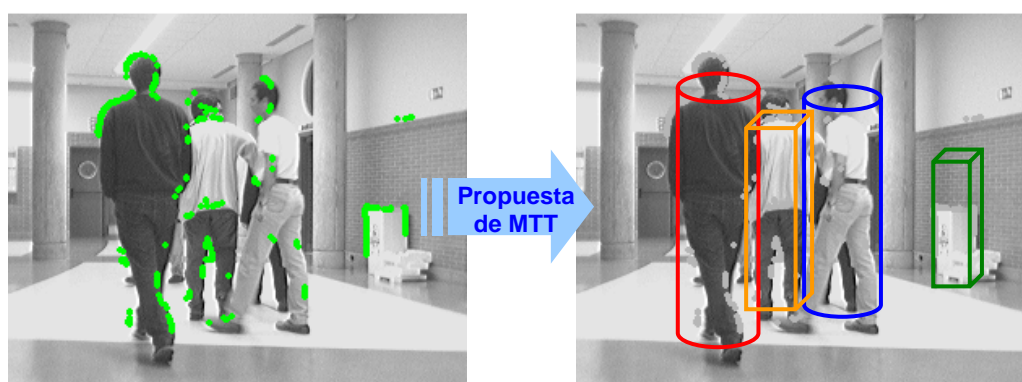


Figura 1.7. Imagen de una escena real extraída de uno de los experimentos que se usan para validar la propuesta de MTT en la tesis. Sobre la imagen izquierda se marcan en color verde a modo de ejemplo el set de puntos a usar como vector de medidas de posición por el estimador propuesto. A la derecha se muestra la misma imagen sobre la que se ha dibujado la que debe ser la salida del proceso de MTT propuesto: un modelo prismático para cada uno de los objetos seguidos que indique su existencia y posición en el entorno.

Tal y como se puede observar en la imagen izquierda de la Figura 1.7, debido a la incertidumbre propia del sistema de adquisición (en el caso de la visión la calibración, las condiciones de luz, etc.), no se logra una información completa del contorno de todos los objetos e incluso la que sí se extrae no es en todos los casos exacta. Así, por ejemplo, algunos como la papelera a la derecha o la persona vestida de negro a la izquierda de la imagen, son mejor detectados que otros, como las dos personas del centro. Además aparecen algunos puntos que no proceden de ninguno de los objetos a seguir en la escena encima de la papelera.

El sistema de MTT ideado en la tesis ha de generar, a partir de este conjunto de datos, un modelo de posición y movimiento, perfectamente identificado, para cada uno de los objetos del entorno de los que se han extraído la información de posición, tal y como se muestra en la imagen derecha de la Figura 1.7. La forma, tamaño y características de este modelo de salida son analizados y elegidos de forma justificada a lo largo de este documento.

En línea con esta especificación de generalidad, no es un objetivo de la tesis la búsqueda de una solución óptima para la extracción de la información de posición del entorno, sobre la que se desarrolle la tarea de MTT. Sin embargo, para poder realizar los ensayos que validen las propuestas presentadas en la tesis se han desarrollado dos sistemas de detección de objetos basados en cámaras que permite obtener conjuntos más o menos densos de datos de posición sobre los objetos en la escena, como el mostrado en la imagen izquierda de la Figura 1.7.

Concretamente, se ha implementado específicamente para la tesis un sistema de visión estéreo basado en dos cámaras estáticamente ubicadas a bordo de un robot que obtiene los puntos de contorno de los objetos encontrados en la escena que no forman parte de la estructura del entorno, y calcula su posición relativa al robot. El funcionamiento de este sistema se describe en detalle en [Marrón06a] y [Marrón07a].

Por otro lado, y con objeto de validar la flexibilidad del algoritmo de seguimiento diseñado en la tesis, en el capítulo 6 éste se prueba también sobre un sistema de adquisición basado en homografías desarrollado de forma paralela a la tesis. El funcionamiento de este otro sistema de observación se describe en detalle en [Pizarro08].

La elección de estos dos sistemas de observación para validar la funcionalidad del algoritmo de MTT propuesto se debe a que representan las dos líneas de investigación en las que, desde hace unos años, se encuentra dividida la comunidad científica en lo que se refiere a la ubicación de los sensores que se utilizan para la navegación autónoma de robots móviles:

1. La política que se viene usando desde la creación de los primeros robots de servicio, consiste en *instalar los sensores a bordo del robot (política de observación centralizada)*. Este método presenta la ventaja de no implicar ningún requerimiento del medio en el que se mueve el robot, abaratando, a primera vista, los costes del proyecto, y haciendo al robot totalmente autónomo e independiente del entorno en el que se encuentre. La contrapartida de esta autonomía está en la necesidad de proporcionar explícitamente un mapa del entorno al robot si se desea localizarlo globalmente dentro de éste. Por otro lado, la perspectiva del entorno que se obtiene a través de los sensores montados en el robot suele ser bastante

pobre, por lo que el proceso de seguimiento de objetos se complica en gran medida, como se demuestra en esta tesis.

2. La política opuesta es la de *ubicar los sensores en el propio entorno de movimiento del robot*, dando lugar a lo que ya se conoce técnicamente como “*espacios inteligentes*” (“*intelligent spaces*” o “*smart rooms*”, [Lee99]) (*política de observación distribuida*). Esta idea surge ante la tendencia cada vez más frecuente de incorporar distintos elementos de control de seguridad (cámaras, detectores de presencia por señal infrarroja, etc.) y redes de comunicaciones (con y sin cable) en las nuevas construcciones. El uso de estos sensores para aplicaciones robóticas abarata el precio final del proyecto, y, por otro lado, permite contar desde un principio con el mapa global del entorno de movimiento del robot y llevar a cabo, por tanto, tareas de localización y seguimiento global sólo a partir de la información extraída de estos sensores. De hecho, estos sistemas de sensado proporcionan información de posición relativa al entorno (absoluta en lugar de la relativa obtenida con la política de sensado centralizada antes descrita). Además, esta política permite incluir cierta inteligencia en los sensores, facilitando la construcción de una arquitectura distribuida que agilice los procesos implicados en la navegación autónoma y permita la compartición de la información entre varios robots que da lugar a la implementación de sistemas multi-robot. La información de posición obtenida de los objetos en el entorno es, en este caso por tanto, mucho más densa y rica que la obtenida el caso anterior, por lo que la tarea de MTT de interés en la tesis se simplifica en gran medida, como se demuestra en este documento.

En esta tesis, y en línea con la idea ya comentada de generalizar la aplicación de los algoritmos propuestos, se opta por realizar pruebas en ambos escenarios, demostrando así la utilidad del sistema de seguimiento diseñado independientemente de la política sensorial usada.

Sin embargo, el desarrollo expuesto en toda la tesis se basa en el uso de la política centralizada de sensado debido que presenta mayor complejidad en la tarea de MTT de interés que la distribuida. Sólo en el capítulo 6 se presenta el sistema de observación distribuido diseñado para validar la propuesta de seguimiento multi-objeto planteada en la tesis. En ese mismo capítulo se incluye un análisis de las implicaciones que la elección de una u otra alternativa de observación tiene en la funcionalidad del sistema de seguimiento propuesto.

1.2. Objetivos y especificaciones de la tesis

Como se comenta en párrafos anteriores, la finalidad general de esta tesis es la de obtener una solución para estimar y seguir la posición de los elementos que rodean a un robot personal en su movimiento por el entorno, constituyendo, de este modo, una aportación en la línea de desarrollo aún sin solución de la aplicación más importante de la robótica, la personal. El objetivo global así descrito se divide en otros más sencillos y en un conjunto de especificaciones que se enumeran en este apartado. Además, para tener un conocimiento más concreto de lo que implica la aplicación de MTT propuesta, en el punto siguiente se incluye una descripción funcional de ésta.

Las condiciones de entorno consideradas (desconocido, complejo, dinámico y poblado por un número múltiple y variable de objetos) así como el uso para la resolución del problema de MTT de una combinación de métodos probabilísticos (que permitan considerar la incertidumbre en la información extraída del entorno y en el comportamiento de los objetos y del robot) y determinísticos (que aporten robustez y velocidad al algoritmo), incrementan el valor científico de los algoritmos aportados en la tesis.

1.2.1. Descripción funcional del sistema de seguimiento objetivo de la tesis

Tal y como se ha comentado hasta este momento, el proceso de seguimiento de múltiples objetos es un problema complicado que no ha sido resuelto aún de forma general por la comunidad científica. Desde un punto de vista práctico, la razón de esta complejidad está directamente relacionada con las difíciles características de entorno implícitas, y radica en el hecho de que la tarea de MTT implica dos tipos de procesos interrelacionados [Hue01]: *la estimación de la posición de cada objeto, el proceso de asociación* de las medidas de posición extraídas del entorno con alguno de los distintos objetos implicados en la tarea de MTT.

1. *En el proceso de estimación*, el modelo elegido para caracterizar el movimiento del objeto a seguir, y las medidas de posición extraídas del entorno, así como el propio algoritmo de estimación son los factores de los que depende fundamentalmente el éxito de la tarea. La elección de modelos y algoritmos que contemplen la incertidumbre implícita en el entorno es imprescindible, tal y como ya se ha comentado, para conseguir un correcto funcionamiento del proceso de estimación.
2. *La asociación o identificación de cada dato de posición con el objeto del que ha sido extraído* es un problema quizás incluso peor resuelto por la comunidad científica para aplicaciones de MTT que el de estimación. Esto se debe fundamentalmente a la carga computacional que conlleva encontrar la solución óptima de asociación, que impide aplicar muchas de las alternativas de asociación propuestas por la comunidad científica al área de la robótica personal. Además es necesario establecer un procedimiento de creación y eliminación de las hipótesis de seguimiento de los distintos objetos, según éstos aparecen o desaparecen del entorno de movimiento del robot, proceso no incluido en muchos de los algoritmos propuestos por la comunidad científica para resolver el problema de MTT.

El algoritmo de asociación debe resolver un conjunto de preguntas que surgen a la hora de relacionar la información de posición obtenida del entorno desconocido con el conjunto de objetos, también desconocido, que se encuentra en él:

- Como el sensor no proporciona información sobre la identidad de la medida este proceso debe obtener este dato con objeto de poder usar el set de medidas correctamente en el proceso de estimación. Sin embargo no se espera del proceso de asociación que, a partir de esta información sin identidad, caracterice con un identificador constante a aquellos objetos que desaparezcan y vuelvan luego a aparecer en entorno del robot.

- Suponiendo cierta incertidumbre o ruido en los modelos de medida, el proceso de asociación debe inferir si la información de posición se ha obtenido de un objeto o de la fuente de error (“*clutter*”⁴) y por tanto se trata de un ruido que hay que filtrar.
- El proceso de asociación debe obtener la solución óptima del problema de asociación en tiempo real, teniendo en cuenta que el número de medidas que se extrae de cada objeto puede variar entre cero y más de una, y el número de objetos a seguir es desconocido.

Muchas de las propuestas planteadas por la comunidad científica para resolver el problema de MTT no tienen en cuenta las premisas anteriores, tal y como se expone en el capítulo siguiente. La solución que se presenta en esta tesis aborda y resuelve todas ellas.

1.2.2. Desglose de objetivos y especificaciones implicadas en el objetivo global

El análisis anterior permite desglosar el objetivo global de la tesis en los siguientes objetivos parciales:

1. *Análisis de distintas alternativas para la extracción de datos del entorno local del robot* a partir de los que se pueda realizar la tarea de seguimiento de los objetos que éstos representan. Como se comenta en párrafos anteriores, se plantea realizar este proceso a partir de un sensor de visión.
2. *Identificación y enumeración de estos objetos*, teniendo en cuenta que se desconoce el número variable de ellos que existen alrededor de robot en cada instante. Además también es objetivo de la tesis la identificación de cada objeto con un único identificador constante a lo largo de su trayectoria continua, sin desaparecer del entorno al margen de los posibles cruces que ésta puede tener con la de otros objetos. Para lograr este objetivo es necesario plantear un algoritmo de asociación que resuelva los problemas de identidad de forma fiable y en tiempo real.
3. *Obtención de un modelo de movimiento* que caracterice individualmente a cada objeto en el entorno, y que incluya variables de posición y velocidad y de ruido asociado a este vector de estado y al sensor de medida de posición con el que se estima el movimiento.
4. *Estimación de la evolución de este vector de estado* en función del movimiento del objeto, mientras se encuentre en el entorno de movimiento del robot. Para realizar esta tarea se propone el uso de una combinación de métodos probabilísticos que tengan en cuenta la incertidumbre incluida en el modelo comentado, y determinísticos que incorporen robustez al algoritmo estocástico de estimación.
5. *Comprobación de la fiabilidad, robustez y flexibilidad* del algoritmo de seguimiento propuesto, frente a la distribución desigual de medidas con las que se realiza el seguimiento,

⁴ En aplicaciones de MTT las medidas de posición que no se corresponden con ningún objeto se agrupan conformando lo que se conoce en lengua inglesa como “*clutter*”, cuya traducción literal al castellano es confusión. Dado que este término está muy extendido en la literatura del área, en este documento se usa junto con el de ruido.

que pueden incluir la presencia de ruidos esporádicos (“*outliers*”⁵) y la oclusión parcial y total de objetos.

6. *Ejecución en tiempo real* del sistema de localización y seguimiento de múltiples objetos en un entorno complejo, que confirme el correcto funcionamiento del sistema propuesto.
7. *Validación de la propuesta de MTT por comparación con otras alternativas planteadas por la comunidad científica.*

Por otro lado, y tras la exposición realizada en este capítulo, la aplicación robótica para la que se abordan estos objetivos implica el cumplimiento de una serie de especificaciones que se enumeran a continuación:

1. *Flexibilidad en entornos no estructurados y desconocidos.* Una de las especificaciones más importantes establecidas por el concepto de robótica personal es la de hacer las aplicaciones diseñadas para este tipo de robots flexibles a su entorno de funcionalidad, independientemente de la complejidad (tipología y tamaño de los objetos), dinamismo (número variable de ellos) o falta de estructura del mismo. La aportación principal de esta tesis se centra justamente en alcanzar esta flexibilidad, frente a otras soluciones desarrolladas por la comunidad científica, que no han tenido el éxito esperado en esta especificación fundamental.
2. *Fiabilidad, robustez y tolerancia a fallos.* El hecho de que los robots personales deban integrarse en entornos complejos dificulta la fiabilidad de sus comportamientos. Por otro lado, la mayor interacción con humanos de este tipo de robots incrementa también la necesidad de establecer especificaciones rígidas de fiabilidad, robustez y tolerancia a fallos a la hora de proponer soluciones en este contexto. La comunidad científica, consciente de este hecho, plantea para la aplicación de MTT de interés el uso de métodos de proceso probabilísticos que permitan adaptarse a errores en el modelo, ruido en las medidas o incluso la ausencia temporal de éstas, debida a oclusiones parciales o totales de los objetos a seguir, permitiendo en todo caso obtener una estimación fiable del estado de los objetos.
3. *Ejecución en tiempo real.* La ejecución en tiempo real de los procesos implicados en la navegación autónoma de los robots personales es imprescindible para que estas máquinas puedan interactuar con el entorno del mismo modo que lo haría un ser humano. Esta especificación implica que el sistema es capaz de procesar la información a la velocidad a la que se capture, y permite estar al corriente de lo que ocurre en cada momento en el entorno. Por este hecho, una de las especificaciones más importantes a la hora de diseñar algoritmos para este tipo de aplicaciones será la de que mantengan un tiempo de ejecución mínimo y constante independientemente de la situación de entorno o de proceso en que se encuentre el sistema robótico en cada momento.
4. *Exportabilidad.* La construcción de un robot capaz de desenvolverse en un entorno cualquiera es el resultado de la unión de distintos módulos inteligentes que resuelven por separado tareas de localización, detección de objetos, reconstrucción del entorno, etc. Esta

⁵ Al dato que no aporta información relevante sino ruido en un algoritmo de procesamiento cualquiera se le denomina “*outlier*” en lengua inglesa. Se trata de un término muy extendido en la literatura científica, por lo que de aquí en adelante se utilizara sin traducir en esta memoria.

tesis no pretende aportar una solución global a todas estas tareas, pero sí a algunas de las más relevantes en el campo de la robótica personal, como es la de seguimiento de múltiples objetos en entornos complejos. La característica de exportabilidad permite un posterior ensamblaje de estos módulos convirtiéndose, por ello, en una especificación fundamental.

El cumplimiento de estas especificaciones se usa a lo largo de toda esta memoria como criterio de valoración cualitativa de los sistemas propuestos para resolver la tarea de interés. Además permiten definir de forma más concreta las aportaciones realizadas en la tesis.

Finalmente, y antes de abordar el contenido de la tesis, merece la pena definir una serie de conceptos que, por su estrecha relación con el tema de interés, deben ser aclarados:

- *Seguimiento (“tracking”)*⁶: Término que se refiere a la tarea de “*estimación de la evolución de la posición y velocidad*” de un objeto cualquiera. Los conceptos de posición y velocidad hacen referencia a los valores relativos respecto al sistema de sensado. Así, el término de tracking se equipara al de localización local en la mayor parte de trabajos de la literatura del área, si bien el segundo término se emplea generalmente en referencia al robot. En este documento se opta por utilizar la palabra seguimiento en lugar de localización para abarcar ambos conceptos.
- *Objeto (“target”)*⁷: Con este término se desea referenciar a todo aquel elemento que el robot encuentre en el entorno desconocido y no estructurado de movimiento y que no forme parte de la estructura arquitectónica del mismo. Los objetos podrán ser, por tanto, estáticos o dinámicos, de distintos tamaños, formas, colores, etc. y animados o no animados; de ahí la importancia de la especificación de flexibilidad impuesta en la tesis.
- *Multimodalidad*: Característica del modelo del sistema de posicionamiento que se desea que el algoritmo de seguimiento sea capaz de manejar, de modo que permita desarrollar el proceso de tracking de un número variable y generalmente múltiple de objetos con un único estimador. Por otro lado, el uso de un estimador que maneje sistemas multimodales permite resolver el problema de recuperación de la posición del objeto a seguir aún cuando la ha perdido (lo que se conoce en tareas de localización como problema de “*kidnapping*” o secuestro). Esta segunda ventaja implícita en el uso de lo que, a partir de este momento, se denominarán estimadores multimodales es más utilizada por la comunidad científica que la primera. El algoritmo que se presenta como solución al problema expuesto en la tesis aprovecha las dos características ventajosas que implica la multimodalidad de un estimador.

1.3. Estructura de la tesis

La organización del libro de tesis es el resultado de una exposición sistemática de la teoría y experimentación de trabajos desarrollados a lo largo del periodo que ha durado la misma.

⁶ Dado que la noción de seguimiento se referencia con el término inglés “*track*” en la literatura científica de la rama, incluida la escrita en lengua española, se propone usar los términos español e inglés por igual.

⁷ El término inglés “*target*” significa, en general, objetivo, y en la literatura científica del área de interés se refiere específicamente al objeto u objetivo de seguimiento. Debido a su amplio uso en referencias bibliográficas del área, tanto en español como en inglés, este término se usará en esta tesis como sinónimo del español.

En el capítulo 2, se realiza un estudio sobre los trabajos realizados por la comunidad científica en el área de interés de la tesis. Se hace especial hincapié en el análisis de los algoritmos, las restricciones y los resultados obtenidos en cada trabajo, así como en el cumplimiento de los objetivos y especificaciones planteados en esta introducción, con objeto de establecer los puntos en los que esta tesis puede aportar distintas mejoras.

A la vista de la revisión del estado de la cuestión expuesto en el capítulo 2, en el capítulo 3 se presenta una descripción general de los distintos algoritmos y procesos desarrollados en la tesis que se pasa a describir en capítulos posteriores, con objeto de alcanzar los objetivos expuestos en la introducción.

Los capítulos 4 y 5 incluyen la descripción teórica y práctica de la implementación probabilística y determinística del algoritmo de seguimiento propuesto. Distintas alternativas y modificaciones de los algoritmos propuestos son analizadas con objeto de que la solución final alcance las especificaciones establecidas en este capítulo primero. En estos capítulos se trata, por tanto, el grueso teórico de la propuesta presentada en esta tesis al problema de interés.

En el capítulo 6 se analiza el sistema de MTT finalmente propuesto y se aplica a distintos experimentos para su análisis y validación práctica, y para evaluar el efecto de sus parámetros en los resultados que genera. Una vez descrito el sistema práctico final, en el mismo capítulo se exponen y analizan los resultados de distintas pruebas realizadas bajo diferentes condiciones de entorno, que permiten validar los objetivos propuestos, bajo las especificaciones establecidas al principio de la tesis.

Además, el algoritmo planteado en la tesis se comparan, en el capítulo 7, con dos soluciones muy extendidas entre la comunidad científica para resolver el problema de MTT de interés, con objeto de validar así el funcionamiento del sistema aquí propuesto.

Finalmente, en el capítulo 8 se concluye esta memoria resumiendo las aportaciones realizadas en la tesis y proponiendo finalmente una serie de líneas de trabajo que permitan aprovechar los desarrollos aportados por la tesis en esta y otras líneas de investigación paralelas.

Al final de la memoria se incluyen una serie de apéndices con documentación complementaria a la expuesta en los capítulos que la componen. En el apéndice I se describen someramente ciertos algoritmos probabilísticos que han sido utilizados en varios de los trabajos analizados en el capítulo 2, pero que no se incluyen en el capítulo 4 porque su funcionalidad se aparta de la elegida en la tesis. Por su parte, en el apéndice II se incluye información complementaria relacionada con la parte determinística de la propuesta de MTT planteada en el capítulo 5 de la tesis. Finalmente, en el apéndice III se describen las pruebas que han sido usadas para hacer un análisis exhaustivo de funcionalidad de los algoritmos descritos en los capítulos 5, 6 y 7.

CAPÍTULO 2

REVISIÓN DE CONOCIMIENTOS

En este capítulo se presenta una revisión de los trabajos más importantes llevados a cabo por la comunidad científica en el área de desarrollo de esta tesis: *el seguimiento*, entendido como *estimación de posición y velocidad, en tiempo real, de un número variable y desconocido a priori de k objetos*, dentro del contexto de la robótica personal.

En línea con el contexto de la tesis, este estado del arte pretende revisar únicamente las aportaciones presentadas con restricciones de diseño semejantes a las que se plantean en esta tesis: *entornos no estructurados, muy poblados, dinámicos y generalmente interiores*, en los que *el sistema de detección de objetos puede encontrarse en movimiento*.

Además, se trata de buscar una solución genérica para el problema de seguimiento de múltiples objetos, frente a las soluciones “*ad. hoc.*” en las que se especifica un entorno o tipo de objeto concreto (especificando el modelo de las medidas, generalmente), de modo que se cumpla la especificación de *flexibilidad* impuesta.

Finalmente, será necesario tener en cuenta las restricciones de observación comentadas también en el capítulo 1, en las que se incluye:

- La posibilidad de que aparezcan “*outliers*” y distribuciones desiguales de medidas de posición debidas a *oclusiones parciales o totales* de los objetos en el entorno de sensado, y
- la *generalidad de modelo de observación* con el que obtener la información de localización de los objetos, si bien teniendo en cuenta el análisis expuesto en el capítulo 1 sobre este

tema, la mayor parte de los trabajos de MTT analizados en este capítulo se basan en medidas de visión.

La existencia de agentes externos no controlables, que interactúan entre ellos y con el robot en su entorno de movimiento, es lo que hace al trabajo abordado en la presente tesis especialmente interesante en el contexto de robótica personal. Las técnicas probabilísticas, ya presentadas en el capítulo anterior, constituyen una herramienta muy adecuada para la resolución del problema de seguimiento expuesto, pues utilizándolas es posible incluir la incertidumbre implícita en los modelos de observación y comportamiento en el estimador, incrementándose con ello la robustez y flexibilidad de los algoritmos.

El uso de métodos probabilísticos facilita, por tanto, la implementación de soluciones genéricas tal y como se pone de manifiesto a lo largo del documento. Por esta razón la mayor parte de las soluciones al problema del seguimiento múltiple que existen en la literatura científica del área de interés se basan en el uso de este tipo de algoritmos. Las propuestas algorítmicas más interesantes de las revisadas en este capítulo son explicadas en el capítulo 4 y el apéndice I, con objeto de compararlas teóricamente con la solución desarrollada en la tesis.

Es necesario destacar que la mayor parte de trabajos de la literatura científica del área de seguimiento se orientan a aplicaciones de vigilancia, basados en modelos de observación bidimensionales que utilizan como sistema de coordenadas el propio plano imagen (espacio UV , imagen izquierda de la Figura 2.1). Los datos así obtenidos son suficientes para seguir objetos en la imagen, pero no en el entorno de movimiento del robot, como se desea en la tesis.

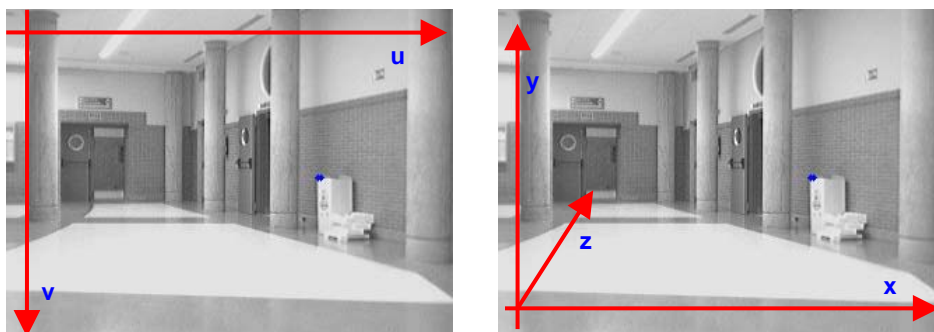


Figura 2.1. Representación sobre una imagen real de los espacios UV y XYZ usados, respectivamente, en las tareas de seguimiento en el plano imagen y en el entorno de movimiento del robot. En esta tesis se resuelve el problema de seguimiento en el entorno del robot, por lo que la información de posición ha de ser tridimensional y estar, por lo tanto, referenciada al espacio XYZ .

Debido a esta razón, la información de posición a partir de la que se implementa el algoritmo de seguimiento múltiple en la tesis ha de ser tridimensional y referida a un sistema de coordenadas cartesiano como el que se muestra en la imagen derecha de la Figura 2.1 (espacio XYZ), en el que el centro de coordenadas lo constituye el sistema de adquisición, esté o no embarcado en el robot.

Para finalizar el capítulo se recogen, a modo de resumen, las principales ideas extraídas del análisis del estado de la cuestión, que permite realizar una comparativa fundamentada de las aportaciones que se realizan en la presente tesis.

2.1. Técnicas empleadas en la tarea de seguimiento de objetos

Antes de abordar la revisión de trabajos conviene realizar un estudio de las distintas técnicas matemáticas planteadas por la comunidad científica para resolver el problema de MTT. Con este objetivo y teniendo en cuenta la división algorítmica de la tarea de seguimiento que se hace en el capítulo 1, la revisión de métodos se organiza en dos partes:

1. Un estudio, desde el punto de vista de teoría de control, de las *distintas técnicas de estimación utilizadas por la comunidad científica en el problema de seguimiento de posición*, con modelos de actuación que incorporen la incertidumbre propia de la aplicación robótica de interés.
2. Otro sobre *técnicas de asociación* en general, y más concretamente, aquellas que por sobre todo son generalmente usadas en la aplicación de tracking de interés.

La combinación de las distintas propuestas de estimación para MTT, con los diferentes algoritmos de asociación, dan lugar a soluciones específicas al problema de seguimiento muy extendidas entre la comunidad científica, como se muestra en el siguiente apartado.

2.1.1. Métodos de estimación

Se plantea en este punto un modelo (de actuación y observación) discreto genérico, muy sencillo y de primer orden, que caracterice la evolución de la posición y velocidad de un objeto cualquiera a seguir, a partir de información simple de posición cartesiana relativa al sistema de adquisición de medidas, en un espacio tridimensional como el que se muestra en la Figura 2.1.

$$\begin{aligned} x_{t+1} &= x_t + v x_t \cdot t_s \\ y_{t+1} &= y_t \\ z_{t+1} &= z_t + v z_t \cdot t_s \end{aligned} \tag{2.1}$$

Las expresiones mostradas en <2.1> definen el comportamiento discreto del modelo planteado, cuya representación matemática en variables de estado se describe mediante las ecuaciones genéricas de estado y de salida mostradas en <2.2>, y se muestra en la Figura 2.2:

$$\begin{aligned} \vec{x}_{t+1} &= A \cdot \vec{x}_t + B \cdot \vec{u}_t \\ \vec{y}_t &= C \cdot \vec{x}_t + D \cdot \vec{u}_t \end{aligned} \tag{2.2}$$

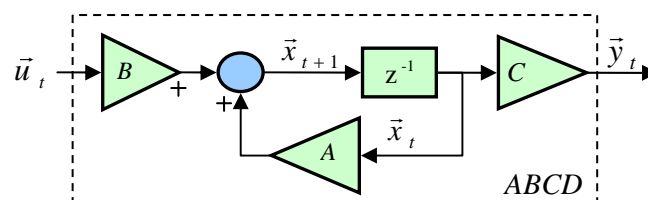


Figura 2.2. Diagrama de bloques del modelo genérico de un sistema expresado en variables de estado. El modelo, que aparece recuadrado en línea discontinua, se representa de más adelante como *ABCD*.

Haciendo $\vec{x}_t = [x_t \ y_t \ z_t \ vx_t \ vz_t]^T$, $\vec{y}_t = [x_t \ y_t \ z_t]^T$ y $\vec{u} = [0 \ 0]^T$, se simplifica la formulación genérica mostrada en <2.2> para obtener un modelo en variables de estado que corresponde específicamente con las ecuaciones <2.1>.

$$\begin{aligned} \vec{x}_{t+1} &= A \cdot \vec{x}_t \\ \vec{y}_t &= C \cdot \vec{x}_t \end{aligned} \quad <2.3>$$

El modelo así descrito se utiliza mucho en aplicaciones de seguimiento como la de interés y recibe el nombre de “*modelo de Velocidad Constante*” (“*Constant Velocity model*”, modelo CV, [Bar-Shalom93]) o también “*modelo de seguimiento de segundo orden*” (“*second order tracking model*”).

En <2.4> y <2.5> se muestran, respectivamente, la especificación concreta de las ecuaciones de estado y de salida para el modelo CV en variables de estado:

$$\vec{x}_{t+1} = \begin{bmatrix} x_{t+1} \\ y_{t+1} \\ z_{t+1} \\ vx_{t+1} \\ vz_{t+1} \end{bmatrix} = A \cdot \vec{x}_t = \begin{bmatrix} 1 & 0 & 0 & t_s & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & t_s \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ z_t \\ vx_t \\ vz_t \end{bmatrix} \quad <2.4>$$

$$\vec{y}_t = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = C \cdot \vec{x}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ z_t \\ vx_t \\ vz_t \end{bmatrix} \quad <2.5>$$

Este modelo permite incluir dentro del vector de estados \vec{x}_t la velocidad del objeto a seguir $[vx_t \ vz_t]^T$, que no es observable, al no poder extraerse del vector de salida del sistema \vec{y}_t , ni obtenerse de la funcionalidad del móvil a seguir, al ser un objeto no controlado (no es un robot) o no especificado, como es el caso en la aplicación de interés.

Para evitar que el modelo del sistema sea únicamente válido para movimiento constante del target a seguir, se introduce un vector de ruido \vec{v}_t en la ecuación de estado (<2.4>) del modelo, que permite caracterizar las variaciones de velocidad del objeto, así como el resto de incertidumbres implícitas en el proceso de estimación:

$$\vec{x}_{t+1} = \begin{bmatrix} x_{t+1} \\ y_{t+1} \\ z_{t+1} \\ vx_{t+1} \\ vz_{t+1} \end{bmatrix} = A \cdot \vec{x}_t + \vec{v}_t = \begin{bmatrix} 1 & 0 & 0 & t_s & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & t_s \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ z_t \\ vx_t \\ vz_t \end{bmatrix} + \begin{bmatrix} v_{x,t} \\ v_{y,t} \\ v_{z,t} \\ v_{vx,t} \\ v_{vz,t} \end{bmatrix}, \quad <2.6>$$

donde $\vec{v}_t = [v_{x,t} \ v_{y,t} \ v_{z,t} \ v_{vx,t} \ v_{vz,t}]^T$ es un vector de variables aleatorias incorreladas entre sí (también conocido como vector de ruido blanco, caracterizado por tener una matriz de covarianza V diagonal) sin ningún otro tipo de especificación estadística, en principio. A la vista de la ecuación <2.6>, la actualización de la velocidad del objeto vendrá dada por:

$$\begin{bmatrix} vx_{t+1} \\ vz_{t+1} \end{bmatrix} = \begin{bmatrix} vx_t \\ vz_t \end{bmatrix} + \begin{bmatrix} v_{vx,t} \\ v_{vz,t} \end{bmatrix} \tag{2.7}$$

Al usar este modelo de movimiento en un proceso de estimación elegido correctamente (que tenga en consideración la incertidumbre propia de éste) para la aplicación de seguimiento, el algoritmo permite obtener, junto con el resto de variables del vector de estado \vec{x}_t , el valor de velocidad del objeto en seguimiento $[vx_t \ vz_t]^T$.

El modelo CV así descrito incluye, además, el factor de incertidumbre necesario para mejorar el comportamiento del sistema de seguimiento en la aplicación de interés. Para completar la definición probabilística del mismo, se añade finalmente al modelo de salida un vector de ruido que \vec{o}_t afecta a las medidas de posición \vec{y}_t .

$$\vec{y}_t = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = C \cdot \vec{x}_t + \vec{o}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ z_t \\ vx_t \\ vz_t \end{bmatrix} + \begin{bmatrix} o_{x,t} \\ o_{y,t} \\ o_{z,t} \end{bmatrix} \tag{2.8}$$

donde $\vec{o}_t = [o_{x,t} \ o_{y,t} \ o_{z,t}]^T$ es también un vector de variables aleatorias incorreladas entre sí (matriz de covarianza O diagonal) sin ningún otro tipo de especificación en principio.

Con estas modificaciones, la forma general del modelo elegido para la estimación de la posición, mostrado gráficamente en la Figura 2.3, queda como sigue:

$$\begin{aligned} \vec{x}_{t+1} &= A \cdot \vec{x}_t + \vec{v}_t \\ \vec{y}_t &= C \cdot \vec{x}_t + \vec{o}_t \end{aligned} \tag{2.9}$$

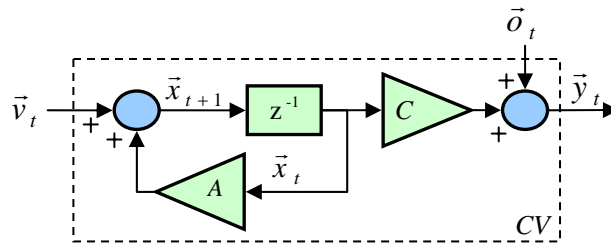


Figura 2.3. Diagrama de bloques del modelo CV en variables de estado, con ruidos de medidas \vec{o}_t y de estado \vec{v}_t .

El modelo CV es uno de los usados más habitualmente en los trabajos de MTT, también llamados de “Seguimiento de Objetos en Movimiento” (“Moving Objects Tracking” MOT) desarrollados por la comunidad científica, junto al “modelo de Aceleración Constante” (“Constant Acceleration model”, modelo CA), que introduce un orden más en el modelo de

movimiento y se usa cuando es posible obtener la velocidad del objeto a seguir por su funcionamiento intrínseco. En ese caso es la aceleración del móvil la que se supone constante y a la que se le añade una señal de ruido que permita modificarla ([Bar-Shalom93]).

El modelo así descrito cumple con la especificación de flexibilidad necesaria para caracterizar el comportamiento de cualquier objeto del entorno independientemente de su tamaño, color, dinámica y tipología.

A continuación se analizan, desde un punto de vista teórico, los distintos algoritmos de estimación utilizados habitualmente por la comunidad científica en aplicaciones de seguimiento como la de interés. A la vista del modelo de posición finalmente elegido para la estimación de la posición de los objetos (expresión <2.9>), será necesario que algoritmo a usar tenga en cuenta el carácter probabilístico del mismo, de modo que se aproveche la incertidumbre incluida para aumentar la robustez del proceso de seguimiento.

Además a la hora de realizar este estudio, se ha tenido especialmente en cuenta si cada una de las técnicas analizadas permite alcanzar la característica de multimodalidad descrita en el capítulo 1. El hecho de que el algoritmo de estimación de posición diseñado en la tesis sea multimodal, es decir, que permita realizar el seguimiento de un número variable y múltiple de k objetos con un único algoritmo, es una de las principales aportaciones de la misma, por lo que los trabajos que cumplan esta característica serán considerados especialmente, con objeto de compararlos con la propuesta diseñada en la tesis.

a) Estimadores clásicos

Los estimadores clásicos, basados en el modelo en variables de estado del sistema a estimar, son los más sencillos. Desde el punto de vista de teoría de control, la tarea de seguimiento consiste en estimar el valor del vector de estado del modelo de posición del objeto a seguir.

La Figura 2.4 muestra el diagrama de funcionamiento del estimador completo en espacio discreto ([Ogata95]), para un sistema con un modelo expresado en variables de estado como el descrito en <2.9> en el que el vector de entrada es nulo.

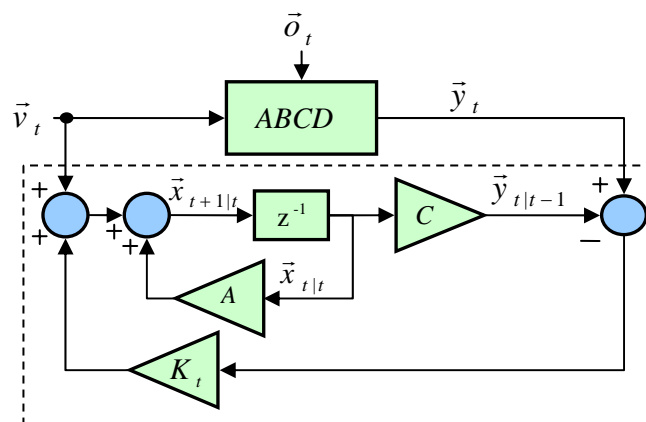


Figura 2.4. Diagrama de bloques de un estimador completo clásico aplicado al modelo descrito en <2.9> y mostrado en la Figura 2.3. El estimador aparece recuadrado en línea discontinua.

El proceso de estimación es recursivo y se puede dividir en dos fases, que se encuentran en todos los tipos de estimadores analizados posteriormente en este apartado:

1. *Predicción* (desarrollada en el instante $t-1$): en la que el modelo del sistema se utiliza para obtener una predicción del estado $\bar{x}_{t|t-1}$.
2. *Corrección* (desarrollada en el instante t): en la que la salida predicha $\bar{y}_{t|t-1}$ se compara con la real \bar{y}_t , y a partir de este error y a través de la matriz de estimación K_t se corrige la evolución del estimador, y se obtiene la estimación corregida del vector de estado $\bar{x}_{t|t}$.

El método de cálculo de la matriz de estimación K_t depende del algoritmo elegido, pero en cualquier caso siempre se define de modo que permita minimizar el error de estimación:

$$error_t = |\bar{x}_t - \bar{x}_{t|t}| \quad \langle 2.10 \rangle$$

Tal y como se desprende del análisis realizado en los párrafos previos, este tipo de estimadores no cumple todos los requisitos definidos anteriormente, pues no es multimodal, por lo que se requeriría uno para cada uno de los objetos a seguir.

Tampoco es robusto ni fiable ante incertidumbres en el modelo, puesto que los vectores de ruido que afectan a las medidas \bar{o}_t y al estado \bar{v}_t no se tienen en cuenta en la estimación del vector de estado¹.

Por todo ello, los estimadores clásicos son sólo útiles para realizar tareas de observación o seguimiento en períodos de tiempo cortos, de modo que el ruido en las señales no haga divergir la estimación, y en las que se conozca el modelo del sistema a seguir de forma exacta.

b) Filtro de Kalman

El “*Filtro de Kalman*” (“*Kalman Filter*”, KF, [Kalman60]) permite obtener la solución que minimiza el error de estimación ($error_t$) a través de la formalización matemática del estimador de estado clásico recursivo, presentado en el apartado anterior.

Tal y como se expone en [Maybeck79], la formulación que da lugar al KF pasa por la presunción de ciertas restricciones en lo que al modelo se refiere:

- *El modelo ha de ser lineal.* En caso de que esta condición no se cumpla, el sistema de ecuaciones lineales <2.2> que caracterizan al modelo genérico en variables de estado pasa a describirse por uno no lineal como el mostrado en la expresión <2.11>. Ante esta situación, se plantea una extensión del KF para resolver el problema de estimación en la que se utiliza una linealización del modelo no lineal obtenida mediante una serie de Taylor. Esta

¹ En la Figura 2.4 el ruido \bar{v}_t que se usa como entrada en el modelo CV sí se tiene en cuenta en la estimación del vector de estado, pero porque se ha introducido de forma controlada. Esto no significa que el estimador clásico tenga en cuenta la incertidumbre del modelo, que al no ser controlada ni conocida no se incluye en el paso de corrección.

extensión del KF se conoce como “*Filtro de Kalman Extendido*” (“*Extended Kalman Filter*” o EKF, [Bar-Shalom93]).

$$\begin{aligned}\bar{x}_t &= f(\bar{x}_{t-1}, \bar{u}_{t-1}, \bar{v}_{t-1}) \\ \bar{y}_t &= h(\bar{x}_t, \bar{u}_t, \bar{o}_t)\end{aligned}\tag{2.11}$$

- Se incorpora incertidumbre al modelo, asociando sendos vectores de ruido al estado \bar{v}_t y a las medidas \bar{o}_t . Los vectores de ruido, deberán estar caracterizados por “*Funciones Densidad de Probabilidad*” (“*Probability Density Functions*”, PDFs) normales y parametrizados por las siguientes condiciones: media nula, independencia (correlación cruzada nula entre los dos ruidos) y, en el caso de ser vectoriales, matrices de covarianza (V_t y O_t) diagonales (vectores de ruido “*blancos*”). La siguiente expresión muestra la expresión matemática que caracteriza a las variables aleatorias mencionadas:

$$\begin{aligned}\bar{v}_t &= \text{norm}(0, V_t) \\ \bar{o}_t &= \text{norm}(0, O_t)\end{aligned}\tag{2.12}$$

El modelo del sistema del que parte el estimador de Kalman se ajusta más a las consideraciones de incertidumbre necesarias en la aplicación de interés. Sin embargo, dadas las restricciones impuestas al modelo, la robustez en la estimación obtenida utilizando un KF sólo se conseguirá para unas condiciones de modelo muy específicas y conocidas, lo que incumple, por otro lado, la especificación de flexibilidad fijada.

Por otro lado, en la formulación completa del KF ([Kalman60]) se observan, al igual que ocurría en el caso del estimador clásico, las dos etapas propias de un estimador recursivo:

1. *Predicción* (desarrollada en el instante $t-1$): en la que se realiza una estimación a priori del vector de estado $\bar{x}_{t|t-1}$ y de la matriz de covarianza del error de estimación $P_{t|t-1}$, mediante la ecuación de estado del modelo.

$$\begin{aligned}\bar{x}_{t|t-1} &= A \cdot \bar{x}_{t-1|t-1} + B \cdot \bar{u}_{t-1} \\ P_{t|t-1} &= A \cdot P_{t-1} \cdot A^T + V_{t-1}\end{aligned}\tag{2.13}$$

2. *Corrección* (desarrollada en el instante t): en la que, a partir del nuevo vector de medida \bar{y}_t , de la ecuación de salida del modelo y de $P_{t|t-1}$, se calcula la matriz de estimación K_t (o matriz de Kalman en este caso) que permite actualizar la estimación del vector de estado $\bar{x}_{t|t}$ y la covarianza de error del error de estimación para ese instante P_t .

$$\begin{aligned}K_t &= P_{t|t-1} (C \cdot P_{t|t-1} \cdot C^T + O_t)^{-1} \\ \bar{x}_{t|t} &= \bar{x}_{t|t-1} + K_t \cdot (\bar{y}_t - C \cdot \bar{x}_{t|t-1}) \\ P_t &= (I - K_t C) \cdot P_{t|t-1}\end{aligned}\tag{2.14}$$

Debido a la formulación del KF, la estimación proporcionada por el filtro $\bar{x}_{t|t}$ presenta un error ($error_t$) que es también una PDF normal y parametrizada:

$$error_t = \left| \bar{x}_t - \bar{x}_{t|t} \right| = norm(0, P_t) \quad \langle 2.15 \rangle$$

El KF es, por tanto, un estimador unimodal, pues si bien el estimador se desarrolla a partir de un modelo basado en PDFs, éstas han de ser normales, es decir, unimodales. Debido a ello el KF no permite resolver el problema de MTT con un único estimador, sino que es necesario usar un filtro por cada target a seguir.

Por todo ello se puede concluir que el KF no es el algoritmo más adecuado para solventar el problema de estimación planteado en esta tesis. Sin embargo la comunidad científica plantea en múltiples trabajos el uso del KF para resolver la tarea de tracking de uno o varios objetos ([Reid79], [Rasmussen01], [Schmitt02], [Zhao04] o [Coraluppi04]) solucionando el problema de la incapacidad del estimador de manejar sistemas multimodales mediante la utilización de un filtro para cada objeto a seguir.

Merece la pena, finalmente, referenciar en este punto al trabajo [Julier97] en el que se plantea una solución mejorada del EKF, denominada “*filtro de Kalman Unscented*” (“*Unscented Kalman Filter*”, UKF). En líneas generales, este algoritmo, que será analizado con mayor detalle en el apéndice I de la tesis, permite conseguir una mejor aproximación de los modelos no lineales implicados en el estimador, que los obtenidos mediante el uso de una serie de Taylor que se emplean en el EKF.

c) Estimadores bayesianos

Tal y como se explica en el capítulo 1, los métodos bayesianos surgidos a mediados de la década de 1990, proponen una formulación alternativa a la de los estimadores clásicos ([Thrun00] o [Arulampalam02]).

La característica principal de este tipo de estimadores es que, al igual que el KF, permiten gestionar la incertidumbre asociada al estado y a las medidas, pero, en este caso, sin las restricciones establecidas por el anterior. Del mismo modo que el KF, el filtro bayesiano propone como salida de estimación una PDF, pero en este caso una genérica en lugar de una gaussiana parametrizada como la que genera el KF ([Arulampalam02] o [Fox03]).

En general, los estimadores bayesianos manejan la información mediante variables aleatorias, caracterizadas por sus respectivas PDFs.

Merece la pena destacar en este punto, que en la literatura matemática se denomina “*función densidad de probabilidad*” a la función continua de probabilidad, mientras que se usa el término “*distribución de probabilidad*” para referenciar a la discretización de la anterior. En esta tesis se usarán indistintamente los dos términos, y se especificará en caso necesario si se trata de una función continua o discreta.

La formalización de un filtro bayesiano pasa, por tanto, por establecer un nuevo modelo para el sistema que en lugar de identificarse por una ecuación de estado y otra de salida, se define por una PDF que caracterice la evolución del vector de estado $p(\bar{x}_t | \bar{x}_{t-1})$ (también conocida

como “*modelo de actuación o de estado*”) y otra para el proceso de medida $p(\bar{y}_t|\bar{x}_t)$ (conocida como “*modelo de percepción o de observación*”).

La salida del estimador bayesiano tampoco es un valor determinístico, sino la PDF asociada a la estimación del vector de estado, a la que se llamará probabilidad de estimación a posteriori o simplemente “*probabilidad a posteriori*” o “*creencia*”.

La salida estimada o creencia representa la probabilidad de que el vector de estado del sistema sea \bar{x}_t , teniendo en cuenta toda la historia del vector de medidas ($\bar{y}_{1:t}$), es decir, la creencia responde a la expresión matemática: $p(\bar{x}_t|\bar{y}_{1:t})$.

El valor de la creencia se obtiene a través de la “*regla de Bayes*” en lo que se denomina “*filtro de Bayes*”, y es este hecho el que le da nombre:

$$p(\bar{x}_t|\bar{y}_{1:t}) = \frac{p(\bar{y}_t|\bar{x}_t, \bar{y}_{1:t-1}) \cdot p(\bar{x}_t|\bar{y}_{1:t-1})}{p(\bar{y}_t|\bar{y}_{1:t-1})} \quad <2.16>$$

La expresión anterior puede simplificarse si se tiene en cuenta la “*condición de Markov*”, que indica que, para cada instante temporal, la historia pasada del sistema puede ser resumida en su estado actual. Considerando que el sistema bajo estudio cumple esta condición, o lo que es equivalente, se comporta como un “*Proceso de Markov*” (“*Markov Process*” MP), la ecuación <2.16> puede simplificarse:

$$p(\bar{x}_t|\bar{y}_{1:t}) = \frac{p(\bar{y}_t|\bar{x}_t) \cdot p(\bar{x}_t|\bar{y}_{1:t-1})}{p(\bar{y}_t|\bar{y}_{1:t-1})} \quad <2.17>$$

Teniendo en cuenta que el conjunto completo de las posibles creencias ha de sumar una unidad, el denominador de la ecuación anterior $p(\bar{y}_t|\bar{y}_{1:t-1})$, que representa la probabilidad total del vector de medidas, suele utilizarse como factor de normalización η , quedando finalmente la ecuación <2.17> como sigue:

$$p(\bar{x}_t|\bar{y}_{1:t}) = \eta \cdot p(\bar{y}_t|\bar{x}_t) \cdot p(\bar{x}_t|\bar{y}_{1:t-1}) \quad <2.18>$$

En la expresión anterior, pueden observarse las tres funciones de probabilidad que caracterizan al filtro bayesiano:

- “*La probabilidad a posteriori*”, o creencia, ya definida: $p(\bar{x}_t|\bar{y}_{1:t})$.
- “*La verosimilitud*”, que viene definida por el modelo de observación del sistema, tal y como se comentó anteriormente: $p(\bar{y}_t|\bar{x}_t)$.
- “*La probabilidad a priori*”, que puede considerarse como una primera aproximación de la creencia: $p(\bar{x}_t|\bar{y}_{1:t-1})$.

A partir de estas definiciones es fácil observar como el filtro de Bayes también se organiza en las dos etapas características de cualquier algoritmo de estimación. Para analizar mejor este hecho se reescribe la fórmula de Bayes según su forma recursiva, que se obtiene desarrollando en la expresión <2.18> el valor de la probabilidad a priori con la “ley de Probabilidad Total”:

$$p(\bar{x}_t | \bar{y}_{1:t}) = \eta \cdot p(\bar{y}_t | \bar{x}_t) \cdot \int p(\bar{x}_t | \bar{x}_{t-1}, \bar{y}_{1:t-1}) \cdot p(\bar{x}_{t-1} | \bar{y}_{1:t-1}) \cdot \delta \bar{x} \quad <2.19>$$

Ecuación que se puede simplificar aplicando de nuevo la condición de Markov:

$$p(\bar{x}_t | \bar{y}_{1:t}) = \eta \cdot p(\bar{y}_t | \bar{x}_t) \cdot \int p(\bar{x}_t | \bar{x}_{t-1}) \cdot p(\bar{x}_{t-1} | \bar{y}_{1:t-1}) \cdot \delta \bar{x} \quad <2.20>$$

La expresión <2.20> se conoce también como “fórmula compacta del filtro de Bayes”, pues en ella se encuentran definidas las dos etapas ya mencionadas en que se basa la funcionalidad recursiva del estimador:

1. *Predicción*: en la que la estimación inicial del vector de estado (probabilidad a priori $p(\bar{x}_t | \bar{y}_{1:t-1})$) se obtiene en el instante t a partir de la creencia obtenida en el instante $t-1$ ($p(\bar{x}_{t-1} | \bar{y}_{1:t-1})$) y del modelo de actuación $p(\bar{x}_t | \bar{x}_{t-1})$ del sistema.

$$p(\bar{x}_t | \bar{y}_{1:t-1}) = \int p(\bar{x}_t | \bar{x}_{t-1}) \cdot p(\bar{x}_{t-1} | \bar{y}_{1:t-1}) \cdot \delta \bar{x} \quad <2.21>$$

2. *Corrección*: en la que se utiliza la probabilidad a priori de la ecuación anterior $p(\bar{x}_t | \bar{y}_{1:t-1})$, el modelo de observación ($p(\bar{y}_t | \bar{x}_t)$ que informa sobre la verosimilitud de las medidas), y la expresión <2.18> del filtro de Bayes para obtener el valor de la creencia $p(\bar{x}_t | \bar{y}_{1:t})$ en el instante t , la salida final del estimador bayesiano.

En este punto, es interesante destacar el hecho de que el KF resulta ser una versión particularizada del estimador bayesiano ([Maybeck79] o [Arulampalam02]), en la que:

- La PDF que caracteriza a modelos de actuación y observación es continua y gaussiana y ha de cumplir las especificaciones ya comentadas.
- La creencia es también una PDF continua y normal, cuya media establece la estimación determinística de salida del KF: $p(\bar{x}_t | \bar{y}_{1:t}) = \text{norm}(\bar{x}_{t|t}, P_t)$.

Existen diferentes versiones del estimador bayesiano en función de la forma que adopten las distintas PDFs implicadas, y es que la aplicación directa de la ecuación compacta de Bayes (más concretamente, de la etapa de predicción del filtro) en el espacio continuo no es implementable al incorporar una integral que debe contemplar todos los valores posibles del vector de estado.

La única implementación aplicada del filtro de Bayes en el espacio continuo es el KF, y ésta implica una simplificación de las PDFs involucradas en el estimador a funciones gaussianas parametrizadas, tal y como ya se ha expuesto anteriormente.

El resto de formas aplicadas del estimador de Bayes se obtienen discretizando alguna de las PDFs implicadas en el mismo. Frente al KF, usado para calcular la solución óptima del proceso

probabilístico a partir de un modelo aproximado del sistema, las versiones discretizadas del filtro de Bayes permiten obtener una solución aproximada usando modelos exactos.

En función de la variable estocástica del proceso de estimación bayesiana que se discretiza se obtienen algoritmos de diferente uso en el área de la robótica:

- En aplicaciones de localización ([Fox99a], [RHINO] o [Trahanias05]), mapeado ([Elfes87] o [MINERVA]), SLAM ([Grisetti05] o [Durrant-Whyte06]) e incluso tracking ([Illman02]), al discretizar el espacio de estado (las distintas posiciones dentro del entorno de interés) mediante un enfoque métrico dando lugar a un enrejillado del mismo, el filtro de Bayes básico se convierte en los denominados métodos de “*Localización de Markov*” (“*Markov Localization*”, ML). En este caso, la creencia $p(\bar{x}_t | \bar{y}_{1:t})$ se representa también mediante un mapa de rejillas de ocupación como el mostrado en la Figura 2.5 (extraído de [Burgard99]).

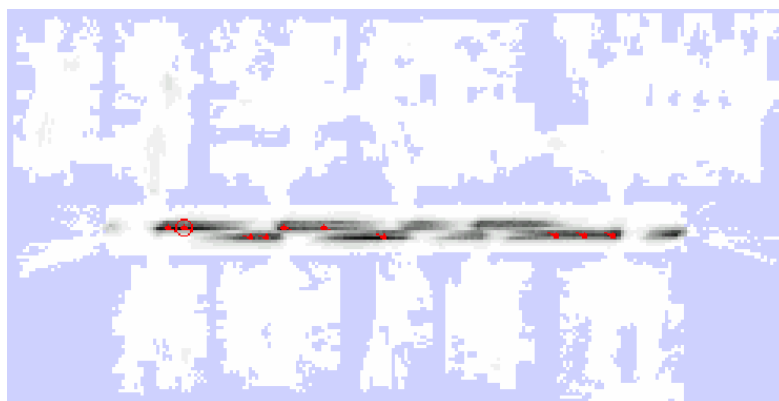


Figura 2.5. Ejemplo funcionamiento de un localizador de Markov sobre un mapa de rejilla de un entorno interior. El mapa se ha obtenido mediante un láser (las zonas ocupadas aparecen en azul, y las zonas libres en blanco) y sobre el se ha dibujado la rejilla de probabilidad que representa la posición en que se encuentra un robot en un instante (en gris). La mayor probabilidad se representa en color más oscuro. Los puntos rojos representan las posiciones más probables del robot, siendo el que aparece con un círculo rojo la posición real. Extraído de [Burgard99].

- En aplicaciones de planificación ([Simmons95], [Pineau03] o [López04]), la discretización del espacio de estado desde un punto de vista topológico en el estimador bayesiano da lugar a los llamados “*Procesos de Decisión de Markov*” (“*Markov Decision Processes*” MDPs) y “*Procesos de Markov Parcialmente Observables*” (“*Partially Observable Markov Decision Processes*” POMDPs). La aplicación de este tipo de algoritmos pasa por realizar una discretización tanto de las PDFs que caracterizan al modelo ($p(\bar{x}_t | \bar{y}_{1:t-1})$ y $p(\bar{y}_t | \bar{x}_t)$), como de la creencia $p(\bar{x}_t | \bar{y}_{1:t})$, obteniéndose representaciones de las mismas mediante grafos y tablas. La Figura 2.6 muestra un ejemplo de aplicación de este tipo de algoritmos a un problema de localización en un mapa topológico (extraído de [López04]).
- Finalmente, en aplicaciones de localización, se utiliza también un filtro de Bayes en el que únicamente se discretiza la creencia $p(\bar{x}_t | \bar{y}_{1:t})$, permaneciendo continuas el resto de PDFs implicadas en el filtro, incluidas las que caracterizan al modelo y al espacio de representación del vector de estado. Como representación de la probabilidad a posteriori

discreta se utiliza un conjunto de n muestras, llamadas “partículas” ($S_t = \{\bar{s}_{i,t} / i=1:n\}$), que son copias del vector de estado \bar{x}_t modificadas por la incertidumbre de la propia creencia y que tienen un peso asociado a cada una de ellas proporcional a su probabilidad individual ($\bar{w}_t = [w_t^{(i)} / i=1:n]$). La ecuación <2.22> muestra la formalización de la creencia así descrita. Esta particularización del filtro de Bayes se denomina “filtro de partículas” (“Particle Filter”, PF), si bien en aplicaciones de localización, como la que se ejemplifica en la Figura 2.7 (extraída de [Doucet01]), se le denomina también “localizador de Monte-Carlo” (“Monte-Carlo Localization”, MCL, [Foxb99b]).

$$\begin{aligned}
 p(\bar{x}_t | \bar{y}_{1:t}) &\cong S_t = \{\bar{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^n \\
 S_t &= \{\bar{s}_{i,t} = \bar{x}_t^{(i)} / i=1:n\} \\
 \bar{w}_t &= [w_t^{(i)} / i=1:n]
 \end{aligned}
 \tag{2.22}$$

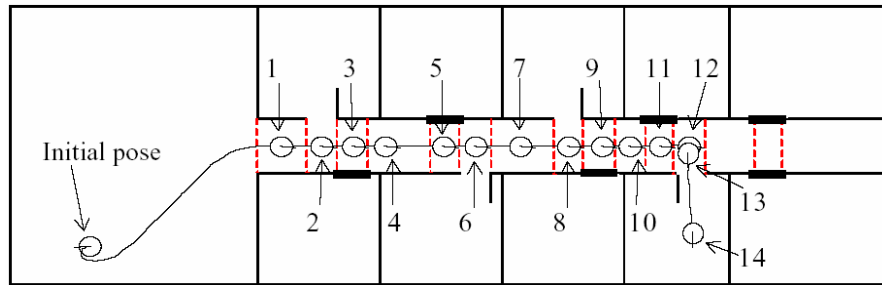


Figura 2.6. Ejemplo de uso de un POMDP para localizar un robot en un entorno mapeado mediante un grafo. La localización del móvil en cada instante se muestra en la imagen mediante un círculo y un número. El número es un identificador que informa del valor discreto más probable de la creencia en cada momento (la posición del robot en el grafo), y los trazos rojos representan la transición entre dos de estos valores. Extraído de [López04].



Figura 2.7. Ejemplo de funcionamiento de un localizador de Monte-Carlo. Las imágenes representan momentos consecutivos del proceso de localización de un robot dentro de un mapa conocido a priori. Los puntos negros (que se van concentrando según se va incrementando la certeza de la posición del robot) representan a las partículas del PF. Extraído de [Doucet01].

El uso de alguna de estas versiones parcialmente discretizadas del estimador de Bayes tiene la ventaja de facilitar el proceso de cálculo en la etapa de predicción del filtro (ecuación <2.21>), que queda, en todos los casos, de la siguiente forma:

$$p(\bar{x}_t | \bar{y}_{1:t-1}) = \sum_{\forall \bar{x}} p(\bar{x}_t^{(i)} | \bar{x}_{t-1}^{(i)}) \cdot p(\bar{x}_{t-1}^{(i)} | \bar{y}_{1:t-1})
 \tag{2.23}$$

Sin embargo, la discretización lleva intrínseca dos desventajas interrelacionadas como se explica más adelante: *ineficiencia en el coste computacional e inexactitud del algoritmo final*.

En el caso del ML, la resolución de las rejillas utilizadas para discretizar las PDFs que caracterizan al modelo y a la creencia, establece el compromiso entre coste computacional y exactitud. Además, al utilizar un espacio métrico de representación, la rejilla generalmente se ve infrutilizada, sobre todo en aplicaciones de localización. Este método es por ello más rentable en aplicaciones de mapeado, en las que se utiliza rejilla completamente para representar el valor estocástico de ocupación de todos los posibles estados (localizaciones) del entorno y una gran exactitud no es tan relevante.

Los métodos basados en representaciones topológicas son más rentables desde el punto de vista del uso de los espacios discretizados, pues en estos casos se incluyen solamente en el espacio de estado los valores de interés para cada aplicación. Sin embargo, estas versiones del estimador de Bayes también adolecen de un problema de coste computacional, pues, al margen de la representación discreta de la creencia, requieren de una representación de los modelos de actuación y de observación basada en tablas que deben ser precalculadas y almacenadas para poder luego ser utilizadas en la ejecución on-line de las dos etapas del filtro de Bayes. Por otro lado, estos métodos se usan generalmente para resolver problemas de planificación, pues no pueden emplearse bajo un enfoque métrico, por lo que no tiene sentido hablar de su exactitud.

El PF es la versión discretizada del estimador de Bayes más rentable desde el punto de vista de la carga computacional. Al discretizar únicamente la creencia, utilizando además una representación en la que sólo aparecen muestras de los valores más probables de la misma, el PF es capaz de presentar mayor exactitud que el resto de alternativas del filtro de Bayes presentadas, en la estimación que arroja como salida, conservando la robustez frente a las incertidumbres propias del modelo, que caracteriza a los estimadores bayesianos, y generando una carga computacional que permite que el algoritmo se ejecute en tiempo real. La ventaja fundamental del PF frente al resto de versiones del filtro de Bayes, que le permite alcanzar estas características, radica en la *no discretización* del espacio de representación del vector de estado.

El filtro de Bayes, en todas las formas presentadas, permite incorporar a los modelos de estado y de medidas la incertidumbre necesaria para facilitar un comportamiento robusto y flexible del mismo ante situaciones complejas, y permite plantear múltiples hipótesis de estimación con un único algoritmo al arrojar como resultado de la estimación una PDF que puede ser multimodal.

Finalizada la revisión de las distintas técnicas de estimación más habituales y aplicables además al problema expuesto en esta tesis, es posible concluir que las basadas en métodos probabilísticos y más concretamente el PF es el único que reúne las condiciones establecidas en la tesis, lo cual permite decidir que este algoritmo es el más adecuado para resolver el problema de estimación multimodal en la aplicación de MTT de interés.

Debido a estas características ventajosas, el PF es, posiblemente, uno de los algoritmos más utilizado por la comunidad científica en tareas de seguimiento de múltiples objetos como la abordada en la tesis. Debido a ello, la revisión de trabajos incluida más adelante, se centra en las

soluciones basadas en PF, que adopta distintos nombres en función del autor, como “*método secuencial de Monte-Carlo*” o “*simulación de Monte-Carlo*” por A. Doucet, C. Hue y J. Vermaak, o algoritmo “*Condensation*” por M. Isard, J. MacCormick y A. Blake, entre otros.

En el capítulo 4 se realiza un análisis teórico más exhaustivo sobre los algoritmos probabilísticos basados en el filtro de Bayes y más concretamente sobre el PF.

2.1.2. Métodos de asociación

Tal y como se explica en el capítulo 1, para implementar un sistema de seguimiento de múltiples objetos, al margen del proceso de estimación de la posición, es necesario definir un proceso de asociación que permita especificar la identidad de cada una de las m medidas incluidas en el set de datos $Y = \{\bar{y}_i / i = 1:m\}$, para saber como aplicarlas al modelo de los k^2 targets a seguir en el algoritmo de estimación de su estado $X = \{\bar{x}_j / j = 1:k\}$. A cada una de estas posibles asociaciones se le denomina hipótesis de asociación $\theta_{i,j} / i = 1:m, j = 1:k$, y a cada solución conjunta de asociación entre m medidas y k targets se le llama evento Φ , y esta formado por una combinación determinada de estas hipótesis $\theta_{i,j}$.

Para tener en cuenta el ruido (outliers) implícito en los procesos de adquisición de medidas, es técnica habitual definir un target más en la lista de los k implicados en el proceso de asociación, con el que se relacionan todos los outliers. A este objeto se le denomina en la literatura inglesa del área “*clutter*”, y sólo se tiene en cuenta en el proceso de asociación, no incluyéndose en el de estimación posterior³.

La asociación de datos es un problema importante y complejo en aplicaciones de MTT; importante porque la asociación de algún dato del set Y con el objeto incorrecto puede hacer divergir la estimación de la posición de ese y el resto de objetos; complejo porque el algoritmo ha de buscar la solución de asociación óptima Φ_{opt} entre todas las posibles, y el número de éstas es función exponencial del número de objetos implicados en el proceso de seguimiento k , y el número de medidas m .

Debido a ello, las propuestas para resolver el problema de asociación son tan variadas como las diseñadas para solucionar el problema de estimación. En este punto se describen funcionalmente los algoritmos de asociación más típicos en el área de MTT ([Bar-Shalom88] o [Rasmussen01]), del mismo modo que en el apartado anterior se hace con los de estimación, analizando las ventajas e inconvenientes que conlleva su uso en la aplicación de interés, en lo se refiere que al cumplimiento de las distintas especificaciones establecidas en esta tesis.

² En este capítulo se va a usar la variable k para caracterizar al número de objetos implicados en el seguimiento, si bien en capítulos posteriores se especificará este nombre, de acuerdo con lo indicado en el glosario de la tesis.

³ En la literatura del área el clutter suele ser referenciado con el subíndice 0 en el listado de targets implicadas en la asociación, por lo que el número total de hipótesis de asociación es proporcional al número de objetos en la tarea de seguimiento k , mas la de subíndice 0, usada para referenciar al clutter.

La revisión de los algoritmos de asociación más típicos para la aplicación de interés, se puede realizar en función de diferentes características de los mismos:

- Del mismo modo que sucede con las propuestas de estimación, se plantean soluciones determinísticas ([Reid79]) y estocásticas ([Fortmann83]) al problema de la asociación. Las segundas permiten considerar cierta incertidumbre en las hipótesis de asociación $\theta_{i,j} / i=1:m, j=0:k$ y calcular un valor de probabilidad combinada $p_{i,j}$ asociado a cada hipótesis $\theta_{i,j}$ en función del resto de las incluidas en cada evento de asociación Φ . Esta probabilidad se usa después en el proceso de estimación del estado de cada target. Los algoritmos determinísticos asignan una certeza absoluta a cada una de las hipótesis de asociación $\theta_{i,j}$ ($p_{i,j} = [0,1]$), y propagan todas las soluciones combinadas en el tiempo. Este hecho los hace impracticables en pocas iteraciones. Por el contrario, los algoritmos probabilísticos obtienen un valor de certidumbre $p_{i,j}$ que puede usarse como acumulador de la verosimilitud de la asociación. Debido a ello, los métodos probabilísticos son más habituales en las propuestas de MTT de comunidad científica, pues su carga computacional es menor que la de los determinísticos, aunque la de estos últimos no implique el cálculo de los valores de probabilidad $p_{1:m,0:k}$.
- En algunas aplicaciones de MTT (en función del tipo de sensor del que proceden los datos de posición, como el RADAR en [Bar-Shalom88]) puede suponerse que el set de datos Y contiene ningún (debido a errores en el sistema de medida o a que el objeto ha desaparecido) o un dato como máximo procedente de cada objeto a seguir. De este modo el número de soluciones del problema de asociación conjunta se reduce a $(k+1)^m$, simplificándose de forma importante la búsqueda de la óptima. Esta premisa es utilizada como base de la mayor parte de algoritmos de asociación que se describen en apartados posteriores, para después plantear alternativas que eliminen esta restricción. De hecho, esta premisa no es aplicable en la tesis, donde se plantea un tratamiento más generalizado de MTT, inicialmente basado en un sensor de visión del que se puede plantear obtener una medida de posición por cada píxel, con lo que el número de medidas extraídas de cada objeto $m_j / j=0:k$ puede resultar alto.
- Con el mismo objetivo de minimizar la carga computacional de la tarea de asociación, la mayor parte de los algoritmos de asociación funcionan bajo la suposición de que el número de targets implicados k es conocido y constante. Como este hecho no se cumple en la mayor parte de aplicaciones de MTT, es habitual generalizar la funcionalidad del algoritmo o diseñar procesos de creación (“*birth*” o “*track initiate*”) y eliminación (“*death*” o “*track deletion*”) de uno o más targets, e incluir su resultado en el algoritmo de asociación básico. Otra técnica también muy usada en aplicaciones de MTT para adaptar la funcionalidad del proceso de asociación a un número k variable de targets, que limita sin embargo el número máximo de objetos a considerar, es la de plantear el proceso para un número constante y suficientemente alto de targets que en principio se encuentran ocultos, e irlos haciendo visibles o invisibles conforme aparecen o desaparecen objetos en la escena de tracking.

Un análisis más general de las características anteriores conduce a la clasificación de los algoritmos de asociación en “*orientados a objetos*” (“*target-oriented*”), que responden a la pregunta de qué target $\bar{x}_j / j=1:k$ asociar a cada medida $\bar{y}_i / i=1:m$, u “*orientados a medidas*” (“*measurement-oriented*”), para responder a la pregunta de qué medida $\bar{y}_i / i=1:m$ asociar a cada target $\bar{x}_j / j=1:k$. Los segundos permiten obtener la solución global del problema de asociación en aplicaciones de número de objetos k desconocido y variable, mientras que los primeros permiten únicamente obtener soluciones subóptimas por lo que requieren la incorporación, como se ha comentado, un proceso específico para modificar el número de targets en la tarea de asociación. Los métodos “*measurement-oriented*” suelen ser, por tanto, determinísticos, ya que generan la solución de asociación óptima, pero tienen un tiempo de ejecución (t_{exe}) mayor.

Debido a lo anterior, los algoritmos “*target-oriented*”, que son en general probabilísticos, son más adecuados para la aplicación de interés, pues aunque no proporcionan una solución óptima al problema de asociación, permiten alcanzar más fácilmente la especificación de tiempo real, difícil de conseguir con los algoritmos que proporcionan la solución global. Al emplear algoritmos “*orientados a objetos*” no es tan necesario, por tanto, establecer restricciones respecto al número máximo de medidas que puede asociarse con cada objeto $m_j / j=0:k$, si bien estas simplificaciones son también bastante habituales en ese caso, pues facilitan la ejecución en tiempo real del algoritmo de asociación.

En función cuales de las características analizadas tenga el proceso de asociación, éste puede complicarse de tal forma que no sea ejecutable en tiempo real para más de dos o tres objetos. Esta es la razón por la que el problema de asociación para MTT sigue sin tener una solución única, y por la que la comunidad científica sigue proponiendo modificaciones a los algoritmos básicos que tienen como objetivo la minimización de su t_{exe} y la maximización de la robustez y la fiabilidad de la solución que generan.

a) “*Seguidor de Múltiples Hipótesis*” (“*Multiple Hypotheses Tracker*”, MHT)

En [Reid79] se propone el algoritmo MHT (“*Multiple Hypotheses Tracker*” o “*Seguidor de Múltiples Hipótesis*”) que genera y actualiza a lo largo del tiempo la solución global de asociación. Se trata, por tanto, de un algoritmo “*orientado a medidas*”.

El algoritmo es determinístico y calcula las $(k+1)^m$ posibles soluciones Φ de asociación que se generan, de forma incremental, en cada instante de tiempo t . Para ello supone que cada medida puede proceder de alguno de los objetos supuestos en $t-1$, de uno nuevo, o del clutter.

El proceso MHT es el único método de asociación que incluye implícitamente la generación de nuevos targets, por lo que está especialmente pensada para aplicaciones de MTT.

Evidentemente, con todas estas características, el conjunto de hipótesis que conforman la solución global generada por este algoritmo se hace inmanejable a las pocas iteraciones del algoritmo, sobre todo si el número de datos m generados en cada instante t , o targets k , es

alto. Debido a ello, el MHT es más habitual en aplicaciones en las que existe una limitación del número de datos a asociar $m_j / j = 0:k$ ya comentado.

En cualquier caso, para facilitar el uso del MHT en aplicaciones reales, se añaden distintos métodos de simplificación de hipótesis al algoritmo básico. De este modo el algoritmo deja de ser global, pero se hace implementable en tiempo real, tal y como se demuestra en [Maksarov95]. Algunas de las técnicas más habituales ([Cox96]) se mencionan a continuación y suelen aplicarse también los distintos algoritmos de asociación que se describen más adelante:

1. *Procesos de eliminación (llamados de “pruning” en la literatura del área) o agrupamiento (llamados de “merging”) de hipótesis* ([Bar-Shalom88]), que permiten eliminar las hipótesis menos probables, o aunar en una sola las más semejantes (que impliquen asociación de más o menos las mismas medidas en más o menos el mismo número de objetos k). El número de hipótesis que se elimina en este paso puede o no estar limitado.
2. *Uso de ventanas temporales para establecer una validez temporal del conjunto de hipótesis* ([Blackman99]), de modo que no se tenga en cuenta toda la historia de medidas $Y_{1:t}$ y targets $X_{1:t}$ en el proceso de asociación sino sólo las implicadas en las hipótesis que siguen manteniéndose en las últimas iteraciones del algoritmo ($t - a:t$), eliminándose el resto. Esta técnica de simplificación puede verse como un “pruning” del conjunto de hipótesis en el espacio del tiempo. Esta técnica se utiliza, no sólo en el MHT, sino en casi cualquier aplicación práctica de los algoritmos de asociación, tanto probabilísticos como determinísticos, pues disminuye de forma importante la carga computacional del proceso, irrealizable en tiempo real en caso contrario.
3. *Proceso de “gating”*⁴ ([Bar-Shalom88]), que se usa para establecer un perímetro de semejanza alrededor de cada objeto en el espacio de definición del proceso de asociación, de modo que sólo los datos que se encuentran dentro del perímetro de gating del estado de cada target $\bar{x}_{1:k}$ son asociados a él. El proceso de gating se usa también para limitar la creación de nuevos objetos a situaciones en las que las medidas estén fuera del gate de todos los ya existentes. Este proceso de simplificación del MHT disminuye también substancialmente su carga computacional, pues, al igual que el siguiente, se ejecuta antes de que las hipótesis $\theta_{1:m,0:k}$ sean calculadas, reduciendo su número, y por ende el de soluciones de asociación Φ , al conjunto $\theta_{0:k} = \theta_{1:m_j,0:k}$, donde m_j referencia al número de medidas que se encuentran dentro del gate asociado al objeto j . En contrapartida, ambos métodos pueden conllevar la pérdida de hipótesis $\theta_{i,j}$ probables, haciendo que el algoritmo sea menos robusto en situaciones complejas de MTT (p. ej. cruces en los caminos seguidos por los objetos o alta incertidumbre en su modelo). Además, en general, la solución de asociación es muy sensible al valor del gate, que suele elegirse en función del nivel de incertidumbre asociado a los modelos de actuación y observación del target.

⁴ El término inglés “gate” referido al problema de asociación tiene difícil traducción, pero en general se usa para definir un límite que permita establecer grupos de relación, y se usa extensamente en la literatura del área, incluso en la de lengua española, por lo que en este documento se emplea sin traducir.

4. *Proceso agrupación (o “clustering”⁵) del conjunto de medidas previo a la asociación*, de modo que las distintas hipótesis $\theta_{i,j}$ se creen entre los $j=0:k_{out}$ targets y los $i=1:k_{in}$ grupos de medidas, disminuyéndose también así de forma relevante, el número de hipótesis de asociación posibles. En este caso, la complejidad está en definir el número de clases k en que organizar el set de medidas Y , del que depende la robustez del algoritmo por la razón expuestas en el párrafo anterior. Un proceso de este tipo, aplicado a un algoritmo de asociación distinto al MHT aquí descrito, se usa en la propuesta de MTT diseñada en esta tesis, como se explica más adelante en la memoria.

El conjunto de eventos de asociación se actualiza a lo largo del tiempo con alguno de los estimadores analizados en el apartado anterior. El KF es el algoritmo más empleado para realizar esta tarea en las soluciones propuestas por la comunidad científica que usan el MHT para resolver la asociación en aplicaciones de MTT ([Reid79], [Maksarov95] o [Schmitt02]).

En estos casos, el estado de cada target $\bar{x}_j / j=1:k$ se actualiza con su modelo de actuación específico en la etapa de predicción del KF:⁶

$$\bar{x}_{j,t|t-1} = A_j \cdot \bar{x}_{j,t-1|t-1} + B_j \cdot \bar{u}_{j,t-1} \quad \langle 2.24 \rangle$$

Una vez realizado el proceso de asociación, el valor de estado predicho se combina, para cada solución de asociación Φ con la o las medidas asociadas $\bar{y}_i / i=1:m_j$ a ésta, en la etapa de corrección del mismo:

$$\bar{x}_{j,t} = \bar{x}_{j,t|t-1} + K_{j,t} \cdot (\bar{y}_{i,t} - C_j \cdot \bar{x}_{j,t|t-1}) \quad \langle 2.25 \rangle$$

En el caso de que una hipótesis contemple la asociación de varias medidas $m_j > 1$ con el target j considerado, será necesario combinarlas (a través de distintos valores estadísticos como su media) para obtener un valor de $\bar{y}_{m_j,t}$ a usar en la ecuación de corrección del vector de estado $\bar{x}_{j,t}$, que queda en este caso, como se muestra a continuación:

$$\bar{x}_{j,t} = \bar{x}_{j,t|t-1} + K_{j,t} \cdot (\bar{y}_{m_j,t} - C_j \cdot \bar{x}_{j,t|t-1}) \quad \langle 2.26 \rangle$$

La PDF resultante del proceso de estimación es una mezcla de gaussianas (puesto que se usa un KF), una por cada solución de asociación creada por el MHT.

En el caso de que el movimiento del target no pueda caracterizarse con un modelo lineal, se usa el EKF (como en [Coraluppi04]) o incluso el PF (como en [Hue02]) como algoritmo de estimación en lugar del KF, con el mismo procedimiento expuesto.

⁵ El verbo “cluster” se traduce al castellano como agrupar, y se incluye aquí sin traducir por su amplio e inequívoco significado en aplicaciones de MTT. En el capítulo 5 se analiza en detalle su significado concreto en la tesis.

⁶ Para simplificar la notación, en este apartado, el subíndice temporal se incluye solo cuando sea necesario para diferenciar una variable de otra con significado distinto.

Este proceso conlleva una importante carga computacional pues ha de ejecutarse por cada solución de asociación generada por el MHT. Debido a ello en todas las referencias que se han analizado en la tesis se usa alguno de los métodos de simplificación expuestos para poder ejecutar el MTT para varios objetos.

b) “*Vecino Más Próximo*” (“*Nearest Neighbour*”, NN)

El algoritmo “*Nearest Neighbour*” (NN, “*Vecino Más Próximo*”) puede verse como una simplificación del MHT en la que sólo se mantiene una hipótesis de asociación $\theta_j / j=0:k$ por cada target, concretamente la que lo relaciona con la o las medidas que se encuentren más próximas a su estado predicho $\bar{x}_{j,t|t-1}$, en el espacio de asociación. Se trata, por tanto, del método de asociación “*orientado a medidas*” que proporciona la solución de asociación Φ que combina, teóricamente, las $k+1$ mejores hipótesis $\theta_{0:k}$ de todas las generadas por MHT.

El NN es un proceso de asociación determinística subóptimo ya que no proporciona una solución de asociación global, sino una única hipótesis por objeto $\theta_{0:k}$, por lo que puede considerarse también como un proceso “*orientado a objetos*”. Este hecho implica la eliminación total de incertidumbre en el proceso de asociación, y la consiguiente disminución de robustez del método ante situaciones complejas, como oclusiones o modelos de movimiento inciertos, muy habituales en aplicaciones de MTT.

Sin embargo, dada a su simplicidad y consiguiente bajo t_{exe} , es un método bastante usado por la comunidad científica en sus propuestas, como se muestra en este documento, ante las siguientes situaciones:

- para problemas de asociación sencillos (targets cuyo estado no interacciona mucho, o basados en un modelo de verosimilitud muy específico),
- o si el número medidas a asociar m es alto, situaciones en las que esta solución es la única que permite conseguir un bajo t_{exe} del sistema de seguimiento.

El NN básico es sólo válido para un número $j=1:k$ de objetos conocido, por lo que al aplicarlo a problemas de MTT se incluye un proceso de gating ([Deriche90]), que permita añadir o eliminar targets en el proceso de asociación, asociando las medidas fuera del perímetro de gate de todos los objetos con el clutter o con un nuevo target. Además, para resolver situaciones en las que una misma medida \bar{y}_i es la más próxima a dos o más objetos en el espacio de asociación, algunas soluciones ([Blackman99]) incorporan un proceso de selección, basado en general en la búsqueda de la organización que minimiza la suma de las distancias entre cada medida \bar{y}_i y el estado del objeto \bar{x}_j al que ésta se asigna.

En esta tesis, el NN se usa en combinación con un algoritmo de clasificación como solución al problema de asociación implícito en la tarea de seguimiento. Esta elección se justifica en el hecho ya comentado de que la combinación del carácter probabilístico del estimador y el determinístico del algoritmo de clasificación permite alcanzar la flexibilidad, robustez y

fiabilidad deseada para la propuesta de MTT, manteniendo su carga computacional suficientemente baja, gracias a la sencillez del NN, como para poder ejecutarlo en tiempo real para un número alto de targets k .

c) “Asociación de Datos Probabilística” (“Probabilistic Data Association”, PDA)

El método de “Asociación Probabilística de Datos” (“Probabilistic Data Association”, PDA, [Fortmann 83]) es un algoritmo, evidentemente probabilístico, que consiste en obtener la probabilidad de que cada medida $i = 1:m$ esté asociada con un único objeto \bar{x}_1 o, por el contrario, proceda del clutter ($p_{i,j} / j = 0,1$).

Se trata, por tanto, de un método de asociación “orientado a objetos”, al igual que el NN y contrario que el método MHT, que genera 2 hipótesis de asociación⁷, una para el target de interés θ_1 y otra para el clutter θ_0 , caracterizadas cada una de ellas, por un set de m valores de probabilidad $p_{1:m,0:1}$ que se obtienen de la combinación de todas las soluciones de asociación.

Por si sólo, por tanto, el PDA no es útil en aplicaciones de MTT, pues sólo permite obtener la probabilidad de asociación del set de datos Y con un único target. La extensión del algoritmo probabilístico a múltiples objetos ($k + 1 > 2$) da lugar al proceso de “Asociación Probabilística de Datos Conjunta” (“Joint Probabilistic Data Association”, JPDA), explicada más adelante.

La incorporación del proceso de gating al PDA es habitual en su aplicación a problemas reales de MTT. Este proceso limita a $ml \leq m$ el número de medidas que pueden asociarse con el target implicado, disminuyendo el número de soluciones de asociación a considerar, y el número de cálculos de probabilidad de asociación a realizar $p_{1:m_1,1}$, pero haciendo el algoritmo, como ya se ha comentado, menos robusto ante situaciones complejas de asociación. Con el mismo objetivo de bajar la carga computacional del PDA, suele ser también frecuente desechar los valores de probabilidad $p_{1:m_1,1}$ más bajos en un proceso de “pruning”.

Finalmente, en aplicaciones prácticas del algoritmo, y como técnica habitual en los métodos probabilísticos, la asociación suele ejecutarse sobre el set de datos instantáneo Y_t , en lugar de usar toda la historia de medidas $Y_{1:t}$, como plantea la solución de asociación óptima.

La aplicación del PDA a problemas de seguimiento de un único objeto es la solución teórica más extendida en esta área ([Bar-Shalom93] o [Rasmussen01]). En general, la propuesta de seguimiento se basa en la combinación de un KF con el PDA para obtener el sistema de

⁷ En muchas referencias consultadas, las hipótesis de asociación no se agrupan para los algoritmos “orientados a objetos”, tal y como se hace en este documento, sino que se mantiene la notación propia de los algoritmos determinísticos “orientados a medidas”, y se plantea una hipótesis por la asociación de cada medida $i = 1:m$ con cada target, coincidiendo entonces el número de hipótesis con el de valores de probabilidad: $(k + 1)^m$. En esta tesis se opta por usar esta notación, con el objetivo de diferenciar las propuestas probabilísticas de asociación, que funcionan “orientadas a objetos” con las determinísticas, que generan soluciones “orientadas a medidas”.

seguimiento buscado, que así constituido recibe el nombre de PDAF “*Probabilistic Data Association Filter*”, es decir “*Filtro de Asociación Probabilística de Datos*”.

El funcionamiento del PDAF es similar al del MHT combinado con el KF, descrito anteriormente. La diferencia radica en que el PDA es un algoritmo “*target-oriented*” y probabilístico, por lo que las medidas $\bar{y}_{1:m_1}$ (suponiendo un proceso de gating) se combinan en la etapa de corrección ponderadas por la probabilidad de asociación $p_{1:m_1,1}$ que genera el PDA a partir de todas las soluciones de asociación posibles:

$$\begin{aligned}\bar{x}_{1,t} &= \bar{x}_{1,t|t-1} + K_{1,t} \cdot \bar{r}_{1,t} \\ \bar{r}_{1,t} &= \sum_{i=1}^{m_1} (p_{i,1} \cdot (\bar{y}_{i,t} - C_1 \cdot \bar{x}_{1,t|t-1}))\end{aligned}\quad <2.27>$$

Al vector $\bar{r}_{1,t}$, que modifica el valor predicho del vector de estado $\bar{x}_{1,t|t-1}$ en la ecuación de anterior para obtener el valor final de estimación $\bar{x}_{1,t}$, se le denomina “*residuo*”.

La etapa de predicción del estimador es idéntica a la expuesta en la expresión <2.24>, puesto que el proceso de asociación no tiene ninguna influencia en ella. Además, la aplicación del PDAF al seguimiento de objetos con modelo de movimiento no lineal, suele realizarse mediante la sustitución de las ecuaciones del KF por las del EKF o del PF, como se explica más adelante.

d) “Asociación Probabilística Conjunta de Datos” (“Joint Probabilistic Data Association”, JPDA)

El método JPDA (“*Joint Probabilistic Data Association*”, “*Asociación Probabilística de Datos Conjunta*”, [Fortmann83]) es una extensión del PDA para aplicaciones de múltiples objetos $k+1 > 2$, como la de interés. En este tipo de aplicaciones el uso de un PDA independiente para cada estimar el estado \bar{x}_j de cada target $j=1:k$ no permite contemplar la interacción de los objetos en la tarea de tracking (p. ej. cruces u oclusiones).

El JPDA surge como la unión de tantos PDAs como objetos estén implicados en el proceso de asociación, a la que se incorpora la probabilidad conjunta de existencia de las distintas hipótesis de asociación. Es decir, cada valor de probabilidad en el JPDA se calcula condicionado al resto, obteniéndose $(k+1)^m$ soluciones de asociación conjunta Φ que definen las $k+1$ hipótesis de asociación $\theta_j / j=0:k$ asociadas a cada target a través de los valores de probabilidad conjunta $p_{1:m,0:k}$, en lugar de a través de los m valores de probabilidad disjunta $p_{1:m,j} = \Lambda_{1:m,j}$ (llamados también valores de verosimilitud, como se explica en el capítulo 6) de cada una de ellas, que daría lugar a k PDAs independientes. De este concepto surge el nombre del algoritmo de asociación JPDA.

Al igual que sucede con el PDA, la definición teórica del JPDA especifica que sólo puede asociarse una medida con cada objeto. Sin embargo, existen soluciones analíticas para

generalizar este algoritmo, como se explica en el capítulo 6. También es bastante usual en las aplicaciones prácticas del JPDA establecer en una ventana temporal de datos a considerar en la asociación de un sólo instante t

El algoritmo básico se define bajo la restricción de que el número de objetos k es conocido, por lo que en este caso también es necesario incorporar un sistema de gating que, además de simplificar al conjunto $p_{1:m_j, j} / j=0:k$ el cálculo de los valores de probabilidad asociados a cada target, permita añadir nuevas hipótesis de objetos al proceso de asociación.

Otro método habitual para disminuir el conjunto de soluciones de asociación y de probabilidades $p_{1:m, 0:k}$ a calcular por el JPDA, es el uso de los procesos de eliminación (“*pruning*”) ya comentados, que desechen las hipótesis de baja probabilidad.

Lo expuesto en los párrafos anteriores permite deducir que el alto t_{exe} es el mayor problema del JPDA, que, sin embargo, se usa recurrentemente en aplicaciones de MTT como la de interés, debido a su gran fiabilidad equiparable a la de la solución global proporcionada por el MHT, que se obtiene a costa de una carga computacional todavía mayor.

La aplicación del JPDA a problemas de MTT da lugar al JPDAF, “*Joint Probabilistic Data Association Filter*”, es decir, “*Filtro de Asociación Probabilística Conjunta de Datos*”, que al igual que el PDAF, descrito anteriormente, surge de la aplicación de las hipótesis de asociación conjunta $\theta_{0:k}$ generadas por el JPDA a k procesos de estimación de la posición individuales (uno para cada target) basados en el KF ([Bar-Shalom88]).

La funcionalidad del JPDAF es, por tanto, idéntica a la del PDAF, sólo que en este caso se implementan k KFs, uno para cada objeto, y en el paso de corrección de cada uno de ellos se utilizan todos valores de probabilidad conjunta $p_{1:m_j, 0:k}$ relacionados con las hipótesis de seguimiento (suponiendo que se realiza un gating de las medidas) para obtener el conjunto de residuos asociados a cada una de ellas $\vec{r}_{1:k, t}$:

$$\begin{aligned} \vec{x}_{j, t|t} &= \vec{x}_{j, t|t-1} + K_{j, t} \cdot \vec{r}_{j, t} \\ \vec{r}_{j, t} &= \sum_{i=1}^{m_j} (p_{i, j} \cdot (\vec{y}_{i, t} - C_j \cdot \vec{x}_{j, t|t-1})) \end{aligned} \quad <2.28>$$

La expresión <2.27> es igual a la <2.28> si se generaliza el PDA para un número múltiple de objetos $j=1:k$. Sin embargo, los valores de probabilidad $p_{i, j}$ no son equivalentes, pues como ya se ha comentado, en el caso del JPDAF se trata de valores de probabilidad conjunta. En el capítulo 6 se describe en detalle esta diferencia.

El JPDAF así definido ha sido usado para resolver el problema de MTT en diferentes trabajos como, de nuevo, [Rasmussen01].

En los últimos años, en [Schulz01] se presenta otro algoritmo de MTT basado en el JPDA, unido, en este caso, a un PF que recibe por ello el nombre de S-JPDAF, “*Sampled JPDAF*” o “*JPDAF Muestreado*” (MC-JPDAF, “*Monte-Carlo JPDA*” en [Vermaak05a], o SIR-MCJPDA, “*Sequential Importance Sampling Monte-Carlo JPDA*” en [Karlsson01]). En el apartado siguiente se comentan algunos de estos trabajos, que tienen especial relevancia en la tesis, tanto por su novedad y rápido éxito entre la comunidad científica como solución al problema de MTT, como por su relación con la propuesta que aquí se presenta.

El origen del S-JPDAF se debe al peligro de divergencia que entraña el uso de un JPDAF en aplicaciones de seguimiento de objetos con un modelo de movimiento muy diferente al gaussiano (p. ej. multimodal). En estos casos, el uso de un EKF o un UKF no asegura la convergencia del estimador. Como ya se ha comentado, el PF es una versión muestreada y no restringida del filtro de Bayes, que permite modelar cualquier tipo de comportamiento, por lo que está especialmente indicado para estos casos.

Para implementar este método de estimación, en general se aumenta el vector de estado para que incluya el de cada uno de los targets implicados en la tarea de MTT $\mathcal{X}_t = \{\bar{x}_{1,t}, \bar{x}_{2,t}, \dots, \bar{x}_{k,t}\} = \bigcup_{j=1}^k \bar{x}_{j,t}$, de modo que cada hipótesis θ_j representada por las partículas del PF S_t implica la existencia de un número concreto de objetos k .

Al igual que ocurre con el JPDAF, la aplicación del S-JPDAF al problema de MTT, conlleva el uso de un PF que actualice todas las hipótesis de estimación. En el S-JPDAF cada partícula del set S_t caracteriza a una solución de estimación diferente, que incluye todas las hipótesis $\theta_{0,k}$ implicadas en el problema de MTT. Además, el peso $w_t^{(i)}$ de la partícula indica la probabilidad de la hipótesis combinada $\theta_{0,k}$ representada por ésta.

La aplicación de un JPDA a un PF se realiza también en el paso de corrección del estimador, usando los valores de probabilidad obtenidos con el algoritmo de asociación $p_{i,j} / i=1:m_j, j=0:k$ para ponderar el valor de verosimilitud $p(\bar{y}_{i,t} | \bar{x}_{j,t})$ asociado a cada medida $\bar{y}_i / i=1:m_j$ en el entorno de gating del target j de interés. La expresión siguiente muestra la forma continua de la ecuación de corrección que se usa en el S-JPDAF:

$$p(\bar{x}_{j,t} | \bar{y}_{1:t}) = \eta \cdot p(\bar{x}_{j,t} | \bar{y}_{1:t-1}) \cdot \sum_{i=1}^{m_j} p_{i,j} \cdot p(\bar{y}_{i,t} | \bar{x}_{j,t}) \quad <2.29>$$

La etapa de predicción del S-JPDAF es la típica de cualquier estimador de Bayes, en la que se usa el modelo de actuación probabilístico de cada target $p(\bar{x}_{j,t} | \bar{x}_{j,t-1})$ para obtener la densidad a priori que lo caracteriza $p(\bar{x}_{j,t} | \bar{y}_{1:t-1})$. La expresión continua de la ecuación que caracteriza a esta etapa de predicción se muestra a continuación:

$$p(\bar{x}_{j,t} | \bar{y}_{1:t-1}) = \int p(\bar{x}_{j,t} | \bar{x}_{j,t-1}) \cdot p(\bar{x}_{j,t-1} | \bar{y}_{1:t-1}) \cdot \partial \bar{x}_j \quad <2.30>$$

En el capítulo 6 se describe también la funcionalidad de este algoritmo de MTT de forma detallada, con el objetivo de comparar su comportamiento con el del propuesto en la tesis ante distintas situaciones y desde el punto de vista de diferentes factores cuantitativos y cualitativos.

El uso de modelos de estimación basados en la expansión del vector de estados $\chi_t = \bigcup_{j=1}^k \bar{x}_{j,t}$ en aplicaciones de MTT es una técnica muy habitual, tal y como se muestra más adelante en este capítulo. Sin embargo, este procedimiento tiene el inconveniente de que su t_{exe} crece exponencialmente con el número de targets k implicadas en el proceso de seguimiento ([Fox03] o [Dellaert03]). De este modo el S-JPDAF y el resto de alternativas basadas en el uso de un único estimador con vector de estado extendido para resolver la tarea de MTT de interés sufre el mismo problema de carga computacional que no permite al JPDAF ejecutarse en tiempo real, en situaciones de más de tres o cuatro objetos. De hecho, el problema es además mayor en el caso de las propuestas basadas en el PF, ya que el t_{exe} de éste supera fácilmente al del KF si se usa un número de partículas n moderado.

Como alternativa al modelo basado en un vector de estado extendido χ_t , en trabajos como [Tweed02], [Vermaak03] o [Khan05] se propone el uso de PF casi independientes para implementar el proceso de estimación. Esta alternativa tiene la ventaja de implicar una menor carga computacional, que además crece linealmente y no exponencialmente con el número de objetos k a seguir, a la vez que mantiene cierta dependencia en la evolución y validación de las hipótesis de seguimiento gracias al JPDA. En muchos de estos casos se sustituye o se añade a este proceso de asociación un modelo de interacción de las targets (como el “*Markov Random Field*” –MRF– o “*Campo Aleatorio de Markov*” propuesto en [Khan05]) que simplifica la tarea de asociación. Más adelante en este capítulo se analizan también estas alternativas en detalle.

En cualquiera de los casos descritos, aunque el uso de un PF como estimador en una aplicación de MTT permite obtener una función de creencia multimodal $p(\bar{x}_{j,t} | \bar{y}_{1:t})$, esta posibilidad no se usa normalmente para caracterizar con una única PDF la posición de un número k de objetos.

En esta tesis se propone usar la multimodalidad de la creencia que permite estimar el PF para que, al combinar su funcionalidad con la de un método de asociación determinístico, cada modo de la creencia represente la posición de un objeto, manteniendo la flexibilidad, fiabilidad y robustez de otras alternativas, y alcanzando la especificación de tiempo real y de t_{exe} constante, que no cumplen ni los algoritmos basados en el JPDA ni los basados en el MHT.

2.2. Trabajos previos en el área de seguimiento de objetos

En este punto se presenta una revisión de los trabajos más interesantes que se encuentran en la literatura científica en el área del seguimiento de múltiples objetos. El estudio constituye un recorrido que comienza con el análisis de las soluciones de tracking de un único objeto (básicas para el desarrollo de las propuestas más desarrolladas) y finaliza con el examen de soluciones

que se resuelven situaciones de seguimiento semejantes a las que se presentan en esta tesis: *un número variable de objetos k , seguidos desde un punto fijo o uno móvil, y que se mueven en entornos complejos muy poblados y de gran dinamismo.*

Como se comenta en apartados anteriores, dos tipos de propuestas han sido planteadas por la comunidad científica a la hora de abordar el problema de MTT:

1. *Utilizar un estimador para cada uno de los k objetos a seguir*, con lo que no se explota la capacidad de modelado múltiple intrínseca a los estimadores bayesianos.
2. *Usar un único estimador y aumentar el vector de estado a estimar* con él de modo que incluya las componentes de estado de cada uno de los k objetos a seguir $(\bar{x}_{1,t}, \bar{x}_{2,t}, \dots, \bar{x}_{k,t})$, dando lugar al vector de estado aumentado χ_t que muestra la siguiente ecuación:

$$\chi_t = [\bar{x}_{1,t} \quad \bar{x}_{2,t} \quad \dots \quad \bar{x}_{k,t}]^T = \bigcup_{i=1}^k \bar{x}_{i,t} \quad <2.31>$$

El valor de la creencia a estimar es en este caso $p(\chi_t | \bar{y}_{1:t})$, y si ésta es multimodal cada uno de sus modos representa una hipótesis de χ_t que incluye tanto el estado de cada objeto seguido $(\bar{x}_{1,t}, \bar{x}_{2,t}, \dots, \bar{x}_{k,t})$, como el número de éstos k .

Ambas soluciones presentan como mayor inconveniente que el t_{exe} del/los estimadores aumenta con el número de objetos a seguir k . Es por ello, que el principal objetivo de los distintos trabajos desarrollados por la comunidad científica en esta área se centra en diseñar versiones del PF que disminuyan su carga computacional, y por ende, le permitan ejecutarse en tiempo real para un número alto k de objetos.

Tal y como se expone en el capítulo 1, en esta tesis se propone utilizar la multimodalidad característica de la PDF implicada en los estimadores bayesianos para modelar con una sola función la creencia del vector de estado de los diferentes objetos a seguir (ver Figura 2.8). A partir de ésta, en este documento se propone identificar distintas hipótesis que representen valores muy probables del vector de estado de los diferentes objetos a seguir $(\bar{x}_{1,t}, \bar{x}_{2,t}, \dots, \bar{x}_{k,t})$, puntos verdes en la figura), y no valores de distinta probabilidad de un único vector de estado $\bar{x}_{1,t}$, como ocurre con la mayor parte de propuestas de la comunidad científica.

Esta alternativa incluye la ventaja del estimador basado en el vector de estados aumentado χ_t de implementar de forma conjunta la estimación de todos los objetos, y mejora la de la carga computacional de la alternativa basada en estimadores independientes, pues tiene un t_{exe} prácticamente constante para un número variable de objetos k .

Por otra parte, la propuesta de estimación multimodal usada en los primeros trabajos de seguimiento basados en el PF, como [Gordon97], presenta importantes problemas de robustez, sobre todo en situaciones de sensado pobre de los objetos a estimar ([Hue02]). El proceso determinístico de clasificación del set de medidas Y_t de entrada al algoritmo de MTT que se

propone combinar en esta tesis con el estimador probabilístico, resuelve el problema de robustez comentado, tal y como se muestra a lo largo de esta memoria. Esta es otra de las aportaciones más relevantes de la tesis.

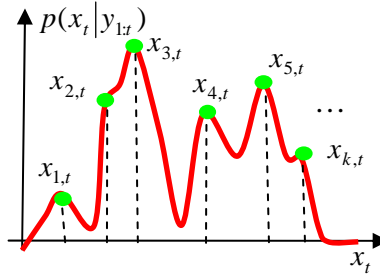


Figura 2.8. Ejemplo de la PDF de una función de creencia $p(x_t | y_{1:t})$ multimodal continua (trazo rojo) de la que se han cogido una serie de muestras (puntos verdes) que pueden representar el valor más probable de la variable x de distintos modelos, en lugar de valores de distinta probabilidad de la misma.

De esta forma, la propuesta planteada constituye una tercera alternativa algorítmica para la resolución del problema de MTT:

3. Utilizar un único estimador multimodal.

El análisis que aquí comienza sólo incluye las propuestas basadas en el PF, pues de lo expuesto en puntos anteriores se extrae que este algoritmo es la mejor alternativa de estimación probabilística para aprovechar el carácter multimodal del estimador frente al problema de MTT.

En la revisión de trabajos se muestran ejemplos de implementación de las tres alternativas de estimación enumeradas, analizando si cumplen las especificaciones establecidas en la tesis, además de las ventajas e inconvenientes y los resultados obtenidos en cada uno de los trabajos, y haciendo hincapié en aquellos que plantean propuestas semejantes a la de la tesis.

2.2.1. Seguimiento de un objeto

El análisis de los trabajos desarrollados por la comunidad científica para el seguimiento de un solo objeto tiene interés en esta tesis, por ser una simplificación del problema de asociación y estimación implícito en esta aplicación, y el punto de partida de propuestas de MTT que se obtienen de la extensión de las anteriores a un número variable de objetos k . Esto se debe a que en general se supone que las medidas extraídas del entorno a partir de las que se realiza el tracking pueden ser ruidosas y proceder del clutter, además del target de interés.

Uno de los primeros trabajos desarrollados por la comunidad científica en el área de seguimiento de objetos con sensores de visión utilizando PFs, es [Isard98a]. Realizado por investigadores de la Universidad de Oxford (Visual Dynamics Research Group), este trabajo presenta el algoritmo “*Condensation*” (apócope de “*Conditional Density Propagation*”, “*propagación de densidad condicional*”), que además es posiblemente el utilizado más veces como referencia en trabajos posteriores. Debido a ello, en este punto se realiza un análisis algo más exhaustivo sobre este trabajo que sobre el resto.

El artículo [Isard98a] incluye una justificación completa de la no viabilidad de algoritmos MHT para resolver el problema planteado, además de una exposición detallada del funcionamiento de un PF, y de las alternativas de implementación del PF que plantean distintos autores en aplicaciones de esta y otras ramas de investigación (un análisis semejante se incluye en el capítulo 4).

El algoritmo “*Condensation*” implementa un seguidor probabilístico para objetos modelados mediante una curva en una imagen. Este tipo de objetos no tienen representación lineal, por lo que no se puede utilizar el KF para realizar su seguimiento. Así se plantea el uso de una PDF discretizada mediante muestras ponderadas, o lo que es lo mismo: *un PF*.

En [Isard98a] el algoritmo “*Condensation*” se pone a prueba en distintas secuencias de imágenes complejas y con varios targets, con resultados satisfactorios. La Figura 2.9 muestra la salida del algoritmo de seguimiento en una de las pruebas. La forma de la curva a seguir se establece manualmente en el inicio del proceso de tracking o mediante un proceso off-line de aprendizaje basado en un “*análisis de las componentes principales*” (PCA “*Principal Components Analysis*”) del objeto de interés, que permiten obtener el modelo de observación.

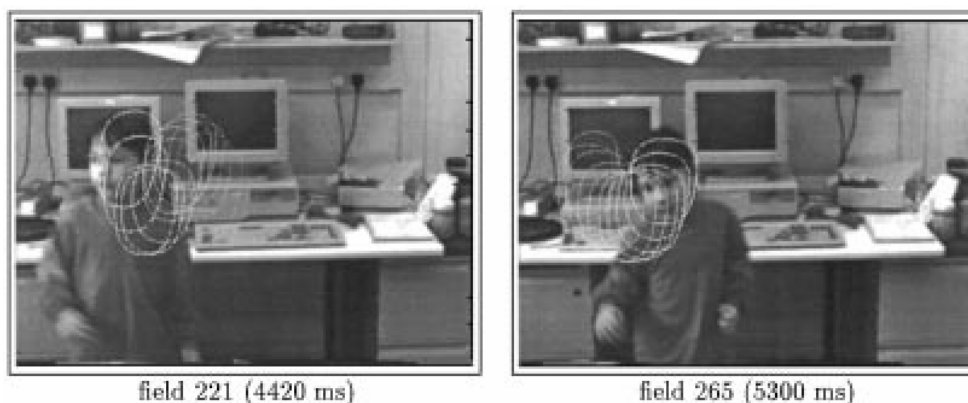


Figura 2.9. Resultados del algoritmo de seguimiento “*Condensation*” ([Isard98a]) ante una secuencia de video de una persona bailando. Las elipses blancas dibujadas en la imagen muestran la evolución de la posición de la cabeza de la bailarina. Se muestra el número de orden de cada imagen y el instante temporal con el que se corresponde.

La prueba mostrada en la Figura 2.9 se ejecuta en tiempo real, a 50FPS⁸ (“*frames/s*” o “*imágenes/s*”) en un procesador a 200MHz, con 100 partículas. Sin embargo, ninguna de las pruebas mostradas en [Isard98a] presenta secuencias con un número variable de objetos k , por lo que el algoritmo “*Condensation*” y los resultados conseguidos con él no son directamente extrapolables a esta tesis.

El modelo específico para el tracking de curvas en imágenes propuesto en [Isard98a] complica el funcionamiento del estimador bayesiano. El uso de modelos, generalmente de observación, definidos para cada aplicación concreta es una de las características más recurridas

⁸ En los resultados temporales mostrados en la tesis se considera tiempo real a un procesado de 15FPS a 20FPS, pues los tests usados se corresponden con un t_s en la captura de imágenes de 15FPS.

en las distintas soluciones presentadas por la comunidad científica al problema del seguimiento de objetos con sensores de visión, tal y como se expone en esta revisión de conocimientos.

Respecto a este tema, como se comenta en puntos anteriores, el objetivo planteado en la tesis, es por el contrario, el de utilizar modelos de actuación y observación genéricos que posibiliten la adaptación de los algoritmos de seguimiento propuestos a distintos tipos de objetos. Trabajos más modernos como [Schulz03a] o [Ng06] apoyan este planteamiento, que, por otro lado, obliga a complicar el algoritmo de estimación y asociación, de forma que se compense la fiabilidad que aporta al seguidor el trabajar con un modelo más específico. El uso de modelos genéricos en el problema de MTT con visión constituye, por tanto, otra aportación de esta tesis a las soluciones presentadas por la comunidad científica hasta el momento.

La aplicación de modelos complejos a un estimador como el PF conlleva un incremento importante de la carga computacional del algoritmo, pues implica la necesidad de usar un número suficiente de partículas que permita obtener una aproximación fiable de cada parámetro del modelo. Los algoritmos resultantes son, en muchos casos, impracticables en tiempo real, lo cuál es una razón añadida para rechazar el uso de este tipo de modelos en la tesis.

Posteriores trabajos de los mismos autores ([Isard98b], [MacCormick00]) se reorientan hacia el área del reconocimiento visual de gestos lo que conlleva una mayor complicación del modelo de observación. Debido a ello se diseñan diferentes algoritmos de seguimiento basados en el algoritmo “*Condensation*”, en los que la novedad radica en idear algún mecanismo que permita al PF gestionar modelos de observación complejos, con un menor coste computacional.

Tal es el caso del algoritmo “*ICondensation*” (“*Importance Sampling Condensation*”)⁹ presentado en [Isard98b], que incorpora al “*Condensation*” ya descrito una función de verosimilitud mejorada, que se basa en el uso de distintas fuentes de medida (color y forma/contorno de los gestos de la mano en [Isard98b]) para mejorar la eficiencia de la etapa de corrección del PF. Con esta combinación se consigue que el seguidor se adapte a los bruscos cambios de dinámica fuertemente no lineal del objeto seguido.

Por otro lado, en [MacCormick00] se propone un PF con “*Muestreo Segmentado*”⁹ (“*Partitioned Sampling*”). En este caso se plantea descomponer la verosimilitud $p(\bar{y}_t | \bar{x}_t)$ en varias funciones con el objetivo de hacer más eficiente el paso de corrección del PF, y por ende el uso de las partículas del algoritmo. Gracias a esta modificación, el PF con muestreo segmentado tiene una fiabilidad semejante al algoritmo básico con la cuarta parte de partículas (según los experimentos mostrados en [MacCormick00] de seguimiento de una mano), lo cuál conlleva una importante reducción del t_{exe} del estimador.

Posteriormente, ambas propuestas son usadas de forma recurrente en otros trabajos, tal y como se detalla a lo largo de este capítulo.

⁹ Los algoritmos “*ICondensation*” y “*Partitioned Sampling PF*” se describe en detalle el capítulo 4 de esta tesis.

Así, años más tarde, un grupo de la Universidad de Illinois en colaboración con investigadores de Microsoft, donde trabajaban en ese momento algunos de los autores de [Isard98a], plantea en [Chen02] otra solución basada en el algoritmo “*ICondensation*”.

El sistema de tracking descrito en ese artículo se pone a prueba en el seguimiento del movimiento de la cabeza de una persona en una escena, en la que ésta es parcialmente ocultada por otra. El modelo de estado del objeto a seguir es también complejo y muy específico (se utiliza una elipse de 30 parámetros), por lo que el estimador, trabajando con 20 partículas, se ejecuta a una velocidad máxima de 10FPS en un procesador a 933MHz. Los resultados de la prueba se observan en la Figura 2.10.

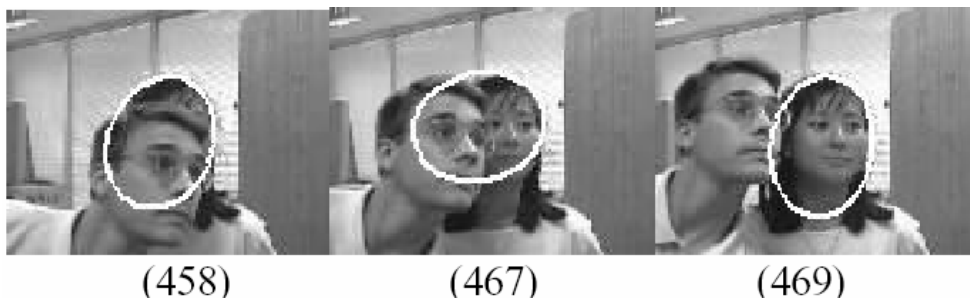


Figura 2.10. Resultados de los trabajos de seguimiento del movimiento de una cabeza humana, extraídos de [Chen02]. Entre paréntesis se muestra el número de orden de cada imagen dentro de la secuencia de video.

El uso del algoritmo “*ICondensation*” en trabajos de seguimiento de caras es muy habitual entre la literatura del área, pues permite combinar modelos de observación sencillos basados generalmente en el color de la piel, con otros más complejos basados en la forma o contorno de la cara. Sirva como ejemplo el trabajo [Chang06], de un grupo de la Universidad de California, en el que se propone un sistema de tracking de caras basado en el “*ICondensation*”. La propuesta añade al PF un modelo de formas y deformaciones de la cara (“*modelos activos de formas o apariencias*”, o “*Active Shapes Models*”, ASMs), aprendidos previamente, que permite reconocer gestos en ella además de seguirla. La especificidad del algoritmo imposibilita su aplicación al problema de interés en la tesis.

Los autores de [Chen02] proponen en [Rui01] otra modificación al algoritmo “*Condensation*” para facilitar, de nuevo, el uso de modelos de observación complejos en el PF, gracias al uso de un UKF. El algoritmo así obtenido es un filtro de partículas “*Unscented*”¹⁰ (“*Unscented Particle Filter*”, UPF, ver [Julier97] o [Merwe01]) que arroja unos resultados similares a los expuestos en [Chen02], y cuyo punto más débil es su alta carga computacional que lo hace difícilmente aplicable a problemas de seguimiento de un número variable y alto de objetos k , tal y como se analiza en capítulos posteriores.

Otra de las propuestas que más éxito a tenido entre la comunidad científica para simplificar la carga computacional del PF es el uso del llamado “*Rao-Blackwellized PF*” (filtro de partículas “*Rao-Blackwellized*”, RBPF¹⁰), en el que sólo parte del vector de estados es estimado a través de un PF.

¹⁰ El UPF y el RBPF se describen en el apéndice I de esta tesis.

Un ejemplo de uso de un RBPF aparece en [Torma03]. En este artículo, un grupo de investigación de la Universidad Eotvos Lorand de Budapest desarrolla un algoritmo de seguimiento por visión de un objeto que se caracteriza mediante una curva “*spline*”, basado en esta modificación del PF. El modelo de observación elegido se divide en una parte determinística y otra estocástica que se usan por separado en el proceso de estimación, de modo que el PF se utiliza únicamente con la parte aleatoria. Además, las dos partes del modelo se combinan después usando la propuesta de “*Importance Sampling*” introducida en [Isard98b].

En la Figura 2.11 se muestran algunos resultados arrojados por el algoritmo propuesto en [Torma03], obtenidos a una frecuencia de 30FPS con un procesador a 1.4GHz. De su análisis se desprende que la variante “*Rao-Blackwellized*” del PF tiene un t_{exe} más corto que el algoritmo “*Condensation*” debido a la importante reducción del número de partículas (de 2000 del PF estándar a 400 en la versión propuesta por los autores) y es, por tanto, más adecuada para aplicaciones con modelos complejos que el segundo.



Figura 2.11. Resultados extraídos de [Torma03] en el seguimiento de un único objeto mediante un RBPF. Los contornos más probables que se obtienen en cada paso de ejecución como salida del seguidor se muestran en negro alrededor del objeto seguido.

De entre los trabajos más recientes desarrollados en el área de seguimiento de un único objeto, y basados en el uso de RBPFs, merece la pena destacar los realizados en el Instituto Tecnológico GATECH, orientados al análisis de la interacción de distintos insectos sociales.

Sirva de ejemplo el trabajo mostrado en [Khan04], en el que se usa este tipo de algoritmo para el tracking de una abeja dentro de un enjambre, tal y como se ve en la Figura 2.12. El seguimiento se realiza a una frecuencia de 15FPS con 500 partículas utilizando un RBPF, mientras que, utilizando un PF estándar, el seguimiento es irrealizable en tiempo real, según los autores del artículo. Evidentemente, para realizar el seguimiento en un entorno tan poblado de objetos de apariencia tan semejante es necesario un modelo de observación de complejo. En este caso se usan 12 componentes de un PCA probabilístico como modelo de color y éstas se organizan en un conjunto lineal y otro no lineal para aplicarse al RBPF.

El algoritmo RBPF aparece también en algunos trabajos de seguimiento de múltiples objetos como [Sarkka07] (donde no se muestran más que resultados de simulación) como mejora a la eficiencia del PF básico en la estimación de su posición.

Otro ejemplo interesante de uso del RBPF en aplicaciones de MTT es el que se muestra en [Schulz03a], donde el PF se usa para obtener, a través de la PDF muestreada $p(\theta_{1:k,t}^{(i)} | \vec{y}_{1:t}) / i = 1:n$, las hipótesis de identidad de un número variable de personas k que se

mueven en un entorno conocido. El PF se usa, por tanto en este trabajo, como algoritmo de asociación probabilística conjunta, JPDA. A partir de esta función de probabilidad muestreada, un conjunto de KFs se usan para estimar la posición de las personas. A pesar de su interés por su peculiar funcionamiento, este algoritmo no presenta el carácter multimodal perseguido en esta tesis para resolver el problema de MTT.

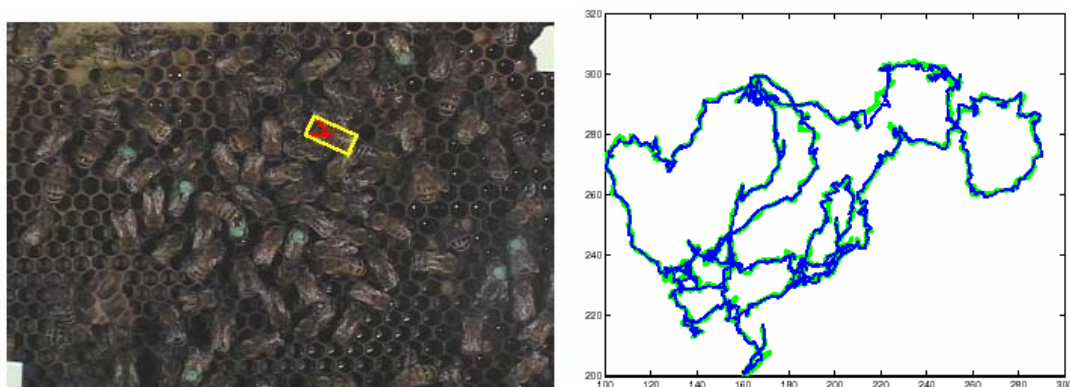


Figura 2.12. Resultados de seguimiento del movimiento de una abeja en un enjambre, presentados en [Khan04]. El movimiento real del insecto, extraído manualmente, se representa en verde, mientras que la salida del estimador probabilístico se presenta en azul.

Al margen de este trabajo, el resto de los expuestos en este punto no describen de forma explícita el método de asociación usado para identificar las medidas que realmente proceden del objeto a seguir del set Y_t extraído del entorno. Haciendo un análisis más profundo se observa que en casi todos los casos se usa un algoritmo NN para elegir, del conjunto de hipotéticas medidas extraídas del target, la más semejante a la predicha según el modelo de observación usado. Esta técnica es suficientemente fiable en estos casos, debido a que sólo existe un posible target con el que asociar las medidas y a que se usan modelos de observación específicos.

Finalmente, en muchos artículos (como [Okuma04] o [Ng06], analizados más adelante) se confirma la idea propuesta en esta tesis de que la organización determinística del conjunto de medidas de entrada Y_t en clases mejora la robustez y disminuye la carga computacional del algoritmo de MTT, facilitando su ejecución en tiempo real.

En esta misma línea, merece la pena destacar el artículo [Sullivan01] de un grupo de investigación de la Universidad KTH en Estocolmo, en colaboración con el grupo de la universidad de Oxford que diseña el algoritmo “*Condensation*”. Este trabajo defiende la idea de robustecer el filtro bayesiano combinándolo con procesos determinísticos, en línea con la propuesta presentada en esta tesis. Sin embargo, el modelo de observación usado es también complejo, basado en la forma y el color del objeto a seguir, por lo que no cumple con la especificación de flexibilidad perseguida en la tesis.

En [Sullivan01] se muestran los resultados arrojados por el algoritmo propuesto en el seguimiento de una persona, y de un balón en un juego de baloncesto. En este caso la inicialización del proceso de seguimiento se realiza de forma manual y sólo para un objeto. En el artículo no se proporciona información sobre la velocidad de ejecución del algoritmo

propuesto, por lo que, teniendo en cuenta además que no se aborda el seguimiento de múltiples objetos, los resultados allí expuestos no son extrapolables a los obtenidos en esta tesis.

En [Odobez04] y [Odobez06], un grupo de investigadores del laboratorio IDIAP suizo, plantean incluir en el modelo de actuación del sistema a seguir, componentes de velocidad que son obtenidas gracias a un proceso de segmentación del conjunto de partículas S_t . La fiabilidad del algoritmo es mayor a la de otros estimadores basados en el PF, tal y como se explica en el artículo. Sin embargo, el clasificador no es usado con los objetivos propuestos en la tesis: se incluye con objeto robustecer el modelo de movimiento del objeto a seguir, y no la multimodalidad del estimador. El algoritmo no se ajusta, por tanto, a los objetivos planteados.

La Figura 2.13 muestra los resultados obtenidos con el algoritmo propuesto en [Odobez06] en el seguimiento de una persona dentro de una habitación. Éstos no son adecuados para un proceso de tracking en tiempo real (4FPS con un procesador a 2.5GHz y utilizando 200 partículas) debido, de nuevo, a la complejidad del modelo de observación usado. En posteriores trabajos, gracias al modelado específico, este grupo de investigación se orienta hacia tareas de tracking multisensorial.



Figura 2.13. Resultados arrojados por el estimador presentado en [Odobez06] para el seguimiento de una persona. En amarillo se muestra la propuesta más probable, en rojo la media de las partículas más probables y en verde la moda de la misma distribución.

Finalmente, en [Kobayashi06] un grupo del laboratorio Sato de la Universidad de Tokio propone un algoritmo de seguimiento de un único objeto basado en un PF y en un detector y clasificador de características en la imagen “Adaboost” ([Viola01]), cuya salida se emplea con el objetivo, propuesto en esta tesis, de robustecer el comportamiento del filtro como estimador.

El artículo muestra resultados obtenidos en tiempo real ($t_{exe} = 30ms$ con 200 partículas) en el seguimiento de un solo objeto (ver Figura 2.14), y se expone como trabajo futuro la ampliación de este algoritmo al tracking de un número variable de targets. En este caso el proceso de asociación se lleva a cabo usando la salida del clasificador de medidas, cuestión que también coincide con lo propuesto en esta tesis, como ya se ha comentado. La propuesta presentada en este artículo tiene, por tanto, especial interés en la tesis, por lo que se analiza más adelante en esta memoria.

Cabe destacar del trabajo expuesto en [Kobayashi06], que el sistema de visión usado para extraer las medidas de posición del objeto a seguir está formado por múltiples cámaras instaladas de forma fija en el entorno, tal y como se observa en la Figura 2.14. Como se

comenta en el capítulo 1, este sistema de sensado descentralizado (exterior al robot personal) propio de los llamados “*espacios inteligentes*”, es cada vez más frecuente en las aplicaciones robóticas por lo que en el capítulo 6 se incluyen pruebas de MTT con el algoritmo propuesto a partir de la información obtenida de esta forma ([Pizarro08]).



Figura 2.14. Resultados del seguimiento de una persona arrojados por el estimador presentado en [Kobayashi06] en un instante de tiempo. Las imágenes muestran la captura de la escena obtenida desde tres de las cámaras usadas como sistema de adquisición. El cuadrado azul muestra la posición estimada del objeto en ellas.

2.2.2. Seguimiento de múltiples objetos con múltiples estimadores

En el centro de investigación de Microsoft se desarrollan también trabajos de tracking de múltiples objetos, más concretamente personas, a partir de información de visión. En [Pérez02], investigadores de Microsoft junto a otros de IRISA en su laboratorio de la Universidad de Rennes, presentan un sistema de seguimiento basado en el algoritmo básico “*Condensation*” que se extiende para su aplicación a secuencias con un número variable de personas k , mediante el uso de un estimador independiente para cada una de ellas.

En este artículo, ampliado en [Pérez04] se propone añadir un modelo de observación basado en el histograma de color de la piel, al expuesto en [Isard98b], basado únicamente en la forma del contorno de la cabeza. El modelo basado en el histograma de color se utiliza en este trabajo, además de para implementar el filtro de Bayes, para detectar la aparición de nuevas personas en la escena. El proceso de detección de nuevos objetos se realiza modelando inicialmente el fondo de la escena mediante su histograma de color, y buscando cambios en ese histograma de fondo que significarán la aparición de un nuevo objeto. El proceso de asociación en este caso se realiza a partir de los modelos de color de cada objeto, es decir, funciona como un algoritmo NN en el espacio de histograma de color. En la Figura 2.15 se observan algunos de los resultados del algoritmo propuesto en [Pérez02] para el seguimiento de múltiples objetos.

El uso de modelos de color en el PF, además de encarecer computacionalmente la ejecución del algoritmo de estimación, tal y como ya se ha comentado, conlleva un decremento de la flexibilidad del estimador, pues lo hace poco robusto ante cambios en las condiciones, principalmente, de luminosidad de entorno. En el artículo no se informa de la velocidad de ejecución del algoritmo para múltiples objetos, si bien se comenta específicamente, que el empleo de histogramas de color es la parte que incrementa en mayor medida el t_{exe} global del seguidor. El sistema propuesto en [Pérez02] no es, por tanto, el más adecuado para llevar a cabo las tareas de seguimiento de múltiples objetos en entornos complejos como el especificado en los objetivos de esta tesis.

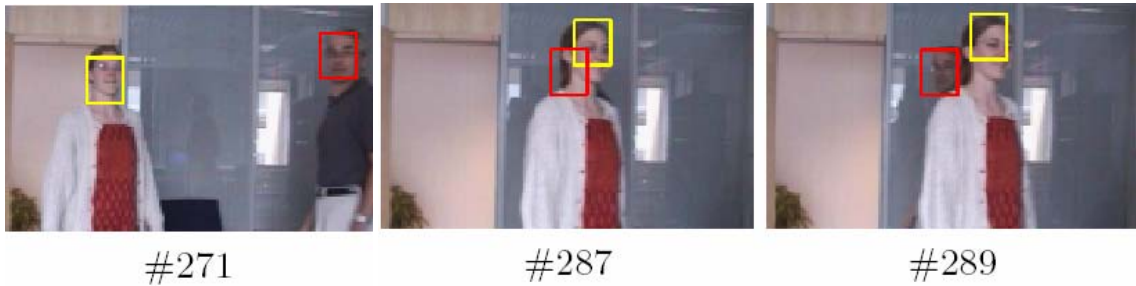


Figura 2.15. Resultados mostrados en [Pérez02] del seguimiento de un número variable de personas k en una secuencia de imágenes obtenidas con cámara estática. Cada objeto viene marcado con un rectángulo de un color diferente. El número de orden de la imagen en la secuencia completa se indica bajo cada una de ellas.

En [Schulz01] (posteriormente ampliado en [Schulz03b]) se presenta una de las propuestas más interesantes planteadas por la comunidad científica para la resolución del problema de MTT basada en el uso de varios PFs casi independientes. Tal y como se explica en el apartado 2.1.2, la aportación de este artículo radica en la inserción de un proceso de asociación de tipo JPDA, en el paso de corrección del PF. Se trata del algoritmo S-JPDAF.

El S-JPDAF se aplica en [Schulz03b] al seguimiento de un número variable de personas (hasta $k=4$ en los experimentos mostrados en el artículo) usando medidas de distancia obtenidas con un láser infrarrojo. Posteriormente esta solución de MTT se utiliza en una aplicación de robótica personal ([Trahanias05], [TOURBOT]). El sistema de seguimiento completo funciona a 5FPS, velocidad algo baja para alcanzar la especificación de tiempo real establecida en la tesis. Este trabajo tiene, por tanto, muchos puntos de conexión con el de interés en la tesis, por lo que se analiza en profundidad más adelante en la memoria.

En el departamento de Computer Science de la Universidad de Bristol se desarrollan también varios trabajos relacionados con PFs aplicados a problemas de seguimiento de un número fijo de objetos k a partir de medidas de visión. En el artículo [Tweed02] se muestran los resultados más importantes que ha obtenido el grupo de investigación en esta aplicación. Allí se plantea un algoritmo denominado “*Subordinated Condensation*” como una modificación del presentado en [Isard98a] para la resolución del problema de MTT.

La idea principal de esta solución consiste en incorporar un modelo de relación entre las partículas que añade restricciones al proceso de asociación de las medidas a los objetos de la imagen, y que permita resolver las oclusiones parciales que se producen en la secuencia de video de forma robusta.

Los autores de [Tweed02] idean un PF organizado en tres niveles jerárquicos, con un número constante de partículas n que se reparte entre los tres niveles independientemente del número de objetos k a seguir en cada momento.

En [Tweed02] se muestra la funcionalidad del algoritmo en el seguimiento de varios pájaros dentro de una bandada en una secuencia de video adquirida con una cámara estática. El modelo de observación utilizado en el estimador se especifica para adaptarse al problema concreto del tracking de pájaros, estableciendo restricciones en la forma y la organización de la estructura física de los animales. Además el algoritmo incluye una etapa de inicialización manual de la

creencia a priori en función de la ubicación inicial de cada pájaro en la imagen. Como se ve en la Figura 2.16, el algoritmo es capaz de seguir más de 25 pájaros de la bandada con 4000 partículas, si bien en el artículo no se especifica que el proceso se realice en tiempo real.

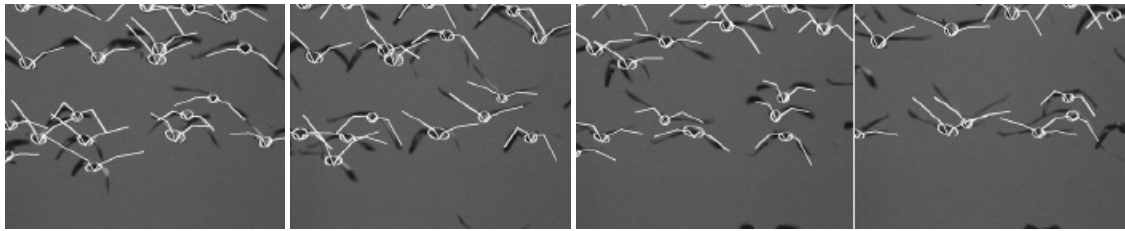


Figura 2.16. Resultados obtenidos con el algoritmo propuesto en [Tweed02] en distintas imágenes de la secuencia de prueba La estructura física de los pájaros cuya posición es seguida por el sistema de tracking propuesto se muestra con trazo blanco.

Es evidente que la solución propuesta en [Tweed02] no es aplicable al problema planteado en esta tesis, pues, por un lado, las estrictas especificaciones de modelo fijadas restringen en gran medida la posibilidad de detectar diferentes tipos de objetos en el entorno de movimiento del robot, y, por otro lado, la carga computacional asociada al algoritmo propuesto no permite conseguir un proceso de seguimiento en tiempo real.

El grupo de investigación del instituto GATECH, cuyos estudios sobre la interacción de animales sociales se comentan en el punto anterior ([Khan04]), desarrolla también algoritmos de tracking de múltiples objetos con información de visión mediante el uso de PFs independientes, como el presentado en [Khan03]. El objetivo de este trabajo, ampliado en [Khan05], es el de llevar a cabo el seguimiento de los objetos existentes en la escena, con una solución de asociación alternativa al JPDA, que conlleve una menor carga computacional y tenga en cuenta la relación social existente entre los objetos seguidos. En líneas generales, la propuesta planteada en [Khan05] consiste en utilizar procesos de estimación diferentes para modelar el movimiento de cada objeto, por un lado, y la relación de posición entre ellos por otro, con una idea semejante la expuesta en [Tweed02].

Para estimar la posición de cada objeto se utiliza un PF que incluye un modelo de asociación de la posición de los distintos objetos en la escena (basado en un MRF) como parte del modelo de observación. Esta información de relación se añade además al vector de estado de los objetos. El MRF caracteriza las relaciones entre dos o más elementos, utilizando para ello un grafo como el mostrado en la Figura 2.17.

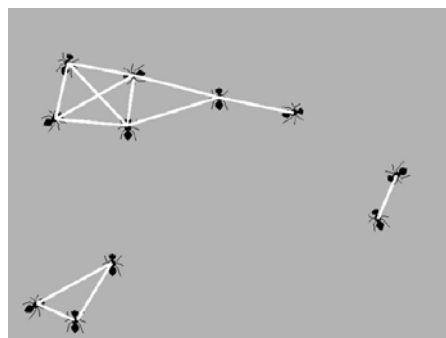


Figura 2.17. Ejemplo del modelo de asociación utilizado en [Khan03] para identificar a los objetos de una escena.

De este modo las restricciones de asociación se usan para incrementar o decrementar la probabilidad de las hipótesis de posición generadas por el PF casi independiente de cada objeto, en lugar de estimar todas las posibles hipótesis mediante un único PF. Los estimadores no son del todo independientes pues la posición estimada de cada objeto j se usa en el pesado de las partículas asociadas a los targets con los que j se relaciona.

Los resultados expuestos en [Khan05] muestran que la solución planteada mejora en más de 150 veces el t_{exe} de un algoritmo JPDAF en una tarea de tracking de dos objetos, y que esta propuesta permite realizar el seguimiento de hasta $k = 20$ objetos, cuestión irresoluble en tiempo real con un JPDAF.

La Figura 2.18 muestra el funcionamiento del algoritmo propuesto (denominado por ellos “*MRF-augmented PF*”), en la tarea de seguimiento de dos hormigas interaccionando entre sí. El t_{exe} de este algoritmo es de unas 15FPS a 30FPS.



Figura 2.18. Resultados obtenidos mediante el algoritmo de MTT presentado en [Khan03]. Los cuadrados verdes marcan la posición y orientación del movimiento (marcado por el triángulo verde interior) de las dos hormigas que interaccionan y son objeto de seguimiento en distintos momentos del experimento.

La característica de multimodalidad implícita en el PF no es, por tanto, aprovechada en la solución propuesta por Khan para desarrollar la tarea de MTT, tal y como se pretende en la presente tesis. Por otro lado, el número de partículas n del PF utilizado por los autores depende del número de objetos k , por lo cual la carga computacional y, consiguientemente, el t_{exe} del estimador multi-objeto es variable. Estos resultados temporales variables no son compatibles con las especificaciones planteadas en los objetivos de la tesis.

2.2.3. Seguimiento de múltiples objetos con un único estimador basado en la extensión del vector de estados

Los autores de [Isard98a] presentan un trabajo en la misma línea de investigación que el anterior, en el que aportan una solución al problema que aparece cuando se producen oclusiones parciales o totales de alguno de los objetos a seguir en la imagen. En el artículo [MacCormick99] se plantea un “*principio de exclusión*” (“*exclusion principle*”) que se incorpora a la PDF de verosimilitud del algoritmo “*Condensation*” y que evita que medidas procedentes de un objeto parcial o totalmente oculto incrementen la creencia de otro diferente.

Para realizar el tracking de múltiples objetos en [MacCormick99] se aumenta el vector de estado incluyendo en él las componentes de todos los elementos a seguir. Además, se incluye en él un identificador indicando si cada objeto oculta a otro. Este modelo de relación, unido al de forma que se usa como verosimilitud en el PF, se emplea en la asociación en caso de oclusión.

El seguidor se implementa, por tanto, mediante una única función de creencia cuya evolución se simplifica gracias al principio de exclusión, disminuyendo el número de partículas n necesarias para su ejecución. Sin embargo, la carga computacional de la solución sigue siendo irrealizable para más de dos objetos, debido al vector de estado extendido y al complejo modelo de observación usados. Como se explica en párrafos anteriores, el muestreo segmentado, introducido en un trabajo posterior de los mismos autores ([MacCormick00]), permite reducir el número de partículas n del PF manteniendo su fiabilidad, lo que resuelve en parte el problema expuesto en [MacCormick99].

Los algoritmos propuestos en [MacCormick99] son validados en diferentes experimentos en los que se lleva a cabo el seguimiento de un número fijo de objetos k , en entornos complejos que incluyen oclusiones totales de estos (ver Figura 2.19). Sin embargo, en ningún caso se aborda la resolución del problema de MTT para un número variable de objetos k .

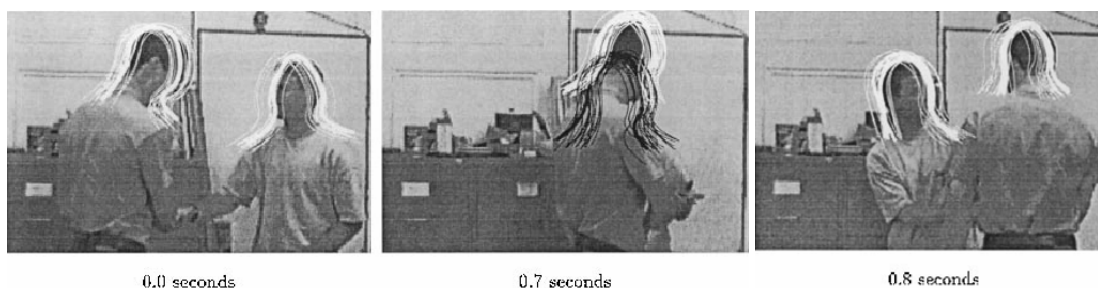


Figura 2.19. Resultados obtenidos en el seguimiento de la posición de dos personas en un entorno estático mediante el uso del algoritmo expuesto en [MacCormick99]. El algoritmo permite realizar el tracking de las dos personas ante situaciones de oclusión parcial y total de una de ellas.

Un equipo de trabajo de la Universidad de California en unión con la corporación Sarnoff, expone en [Tao99] una variación del algoritmo básico de seguimiento de múltiples objetos con un único PF de vector de estado aumentado. La novedad se presenta en la incorporación a la creencia de información adicional sobre la interrelación de los distintos objetos en la escena (de forma semejante a lo expuesto en párrafos anteriores, en relación con trabajos como [Tweed02] o [Khan05]) con objeto de simplificar el número de hipótesis de estimación válidas. El proceso de asociación se hace, por tanto, también en este caso, usando esta información de comportamiento del conjunto de k targets.

En la propuesta presentada en [Tao99] el vector de estado incluye además dos variables que modelan la probabilidad de incorporación y eliminación de objetos del entorno, por lo que el sistema propuesto puede utilizarse en el seguimiento de un número variable de objetos k . Por otro lado, se usa un modelo de observación jerárquico (a nivel de píxel y a nivel de “blob”¹¹) y complejo basado en la forma del objeto.

¹¹ El termino inglés “blob”, cuya traducción directa podría ser mancha, se refiere a una zona de la imagen con unas determinadas características de color, intensidad luminosa o textura. Debido a ser un término utilizado recurrentemente en la literatura científica relacionada con el área de interés se utilizara sin traducir en la tesis.

El algoritmo así obtenido no es capaz de ejecutarse en tiempo real (1FPS en un procesador a 400MHz, ejecutando un PF con 300 partículas), debido a la complejidad del modelo de estado con el que se trabaja. La Figura 2.20 muestra algunos de los resultados expuestos en [Tao99].

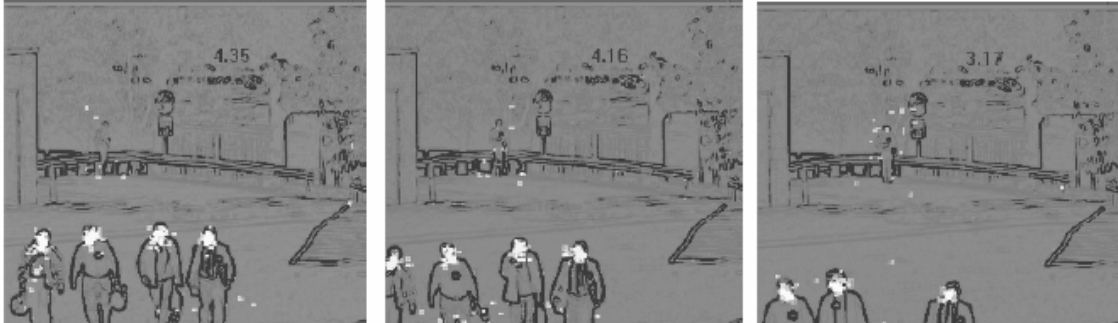


Figura 2.20. Resultados expuestos en [Tao99] en la aplicación del algoritmo presentado al seguimiento de un número múltiple y variable de objetos k . Los puntos blancos de cada imagen en la secuencia completa representan a las partículas del estimador bayesiano.

Mas tarde, en [Spengler03] un grupo de investigadores del Instituto Federal de Tecnología de Zurich (ETH) presenta una idea semejante a la expuesta por Tao en el trabajo anteriormente comentado, como solución al problema de MTT. En este caso se emplea también un modelo de observación basado en la forma de los objetos, que se aprende en un proceso previo off-line.

Aunque en el artículo no se indica el t_{exe} del algoritmo expuesto, el número de partículas utilizadas en los experimentos (alrededor de 4000) hace pensar que el estimador propuesto no puede usarse para llevar a cabo tareas de seguimiento en tiempo real de un número de objetos k mucho mayor al mostrado en el artículo (tres o cuatro objetos).

Los resultados que se exponen en [Spengler03] (ver la Figura 2.21) muestran el seguimiento de varios objetos en una escena capturada con cámara estática.



Figura 2.21. Resultados obtenidos con el algoritmo presentado en [Spengler03] en el seguimiento de dos personas. Las elipses blancas muestran la posición en la imagen del objeto seguido y las líneas blancas indican el estado del modelo de interrelación usado.

Los mismos autores de [Isard98a] plantean otra de las primeras soluciones al tracking de múltiples objetos mediante una única PDF basada en el uso de un vector de estado aumentado. Este trabajo se presenta en [Isard01] y expone, como aportación más interesante, un modelo de observación en base a blobs, obtenidos mediante un entrenamiento off-line del sistema de

detección. El algoritmo así desarrollado recibe el nombre de “*BraMBle*”, y es usado como referencia (sobre todo en lo que al modelo de observación se refiere) en otros muchos trabajos del área, como el descrito anteriormente en [Pérez02], y otros analizados en posteriormente.

La PDF conseguida mediante el mencionado proceso off-line se usa para detectar las personas en la escena y aumentar el vector de estados a estimar introduciendo el de los nuevos objetos a seguir. Además, el trabajo con blobs permite agilizar el proceso de estimación y de asociación, que se realiza con un JPDA. El inconveniente, ya comentado, de este tipo de modelos de observación es que sólo es válido para el seguimiento de un tipo concreto de objetos, lo cual no cumple con la especificación de flexibilidad establecida.

En este trabajo se recurre de nuevo a la técnica de incluir información sobre la interacción entre los distintos objetos de la escena dentro del vector de estado a estimar, aumentando así la robustez del estimador ante oclusiones de los objetos a seguir, y disminuyendo la carga computacional del algoritmo de MTT.

Los resultados presentados en [Isard01] se obtienen a una velocidad de 15FPS en escenas con dos o tres personas en movimiento, ejecutando el algoritmo de seguimiento con 1000 partículas en un procesador a 477MHz (ver Figura 2.22). Sin embargo en el artículo no se exponen resultados del seguidor en escenas más complejas pues el número de partículas n necesarias para obtener una estimación robusta imposibilita estos ensayos.



Figura 2.22. Resultados del sistema de seguimiento planteado en [Isard01]. Los polígonos de color indican la estimación de posición arrojada por el algoritmo en diferentes instantes de la secuencia de prueba global.

Al igual que otras propuestas ya descritas, el algoritmo “*BraMBle*” se revisa posteriormente en esta tesis debido a su éxito entre la comunidad científica para aplicaciones de MTT.

Un ejemplo de uso del algoritmo “*BraMBle*” en trabajos posteriores es el presentado en [Branson05] por un grupo de investigación de la Universidad de California. En este artículo se plantea el seguimiento de un número constante y conocido k de ratones. La complejidad en este caso aparece en las continuas oclusiones entre los objetos a seguir. Este problema se resuelve usando un modelo de observación basado en blobs de color y forma, que, como en [Isard01] se obtienen mediante un proceso de aprendizaje off-line previo. Con este método el algoritmo resuelve el seguimiento en la mayor parte de las situaciones de oclusión, si bien no conserva la identidad de los ratones en muchos de los casos.

El algoritmo mostrado en [Branson05] no es, por tanto, tampoco aplicable al problema de interés en la tesis, pues usa un modelo de observación complejo, con el que no se cuenta en este

caso, que permite obtener un contorno de los ratones. El modelo de observación es tan específico que se usa en la tarea de asociación implícita en el proceso de MTT. Además el algoritmo no es válido para un número variable de targets k .

En la Figura 2.23 se muestra, en ocho imágenes extraídas del artículo (de la a) a la h)), el resultado de seguimiento de dos ratones conseguido con el algoritmo propuesto. En el documento no se informa de la carga computacional del algoritmo, por lo que no se puede asegurar que éste se ejecute en tiempo real, teniendo en cuenta además la complejidad del modelo de observación usado y que se usan 2000 partículas para el seguimiento de tres ratones.

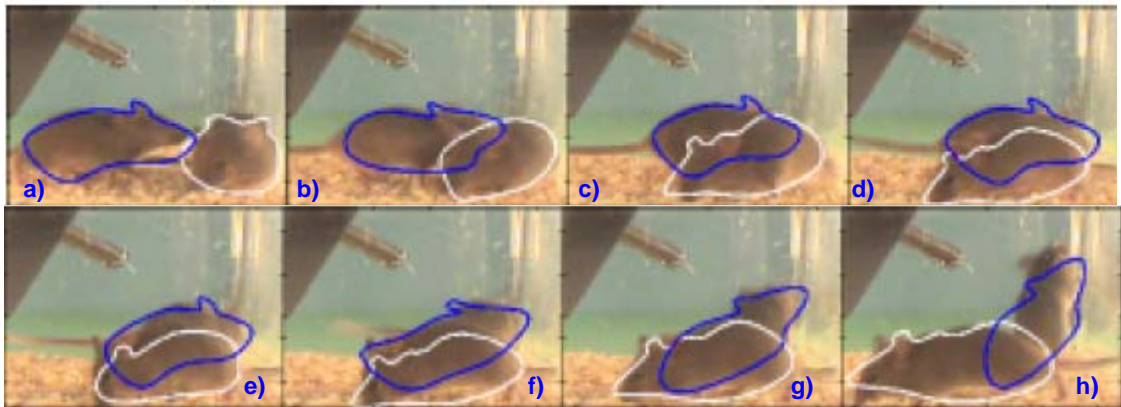


Figura 2.23. Resultados extraídos de [Branson05], en el seguimiento de dos ratones. Se muestran ocho imágenes de la prueba (de la a) a la h)) en las que el modelo de contorno asociado a cada ratón se muestra en un color diferente.

En [Hue01] el grupo de investigadores que desarrollan posteriormente el trabajo [Pérez04] ya analizado, presenta un sistema de seguimiento de un número constante de objetos k basado en un PF con vector de estado aumentado. En este trabajo el modelo de estado incluye una variable que caracteriza la probabilidad que tiene cada nueva medida en ser asociada a cada objeto de la escena. Esta variable se obtiene de una versión probabilística del algoritmo de asociación MHT (el MHT probabilístico o PMHT, “*Probabilistic Multi-Hypothesis Tracker*”), cuya funcionalidad se describe en el punto anterior.

En la Figura 2.24 se muestran los resultados obtenidos por los autores en la aplicación del algoritmo propuesto al seguimiento de tres objetos. Como función de verosimilitud en [Hue01] se utiliza un modelo de contorno de personas aprendido off-line. En el artículo no se especifica que el algoritmo se ejecute en tiempo real, pero su complejidad hace pensar que no sea así.

En posteriores artículos de los mismos autores ([Hue02]) se plantea la extensión del algoritmo propuesto en [Hue01] a un número cambiante de objetos k . Sin embargo en el artículo no se muestran resultados reales del nuevo sistema, aunque sí comparativas exhaustivas de éste con otros ya conocidos como el S-JPDAF.

El mismo grupo de investigación que presenta el artículo [Odobez06] analizado en el punto 2.2.1, propone en [Smith05] otra modificación a la versión básica del PF con vector de estado aumentado para realizar el seguimiento de un número variable y múltiple de objetos k .



Figura 2.24. Resultados extraídos de [Hue01], en la aplicación del algoritmo ahí descrito al seguimiento de tres personas. Se muestran tres imágenes de la secuencia de video de prueba.

En el algoritmo de estimación propuesto por Smith se genera un modelo de observación organizado en dos órdenes jerárquicos: uno a nivel de píxel y otro a nivel de blob de color. En este algoritmo se utiliza también una PDF para modelar la relación de posición entre los distintos objetos de la escena (una versión mejorada del expuesto en [Khan05]), que se usa en el proceso de asociación de la tarea de MTT. En el artículo no se da información al respecto de la carga computacional del algoritmo final, pero a partir de su descripción funcional es lógico esperar que ésta sea alta. La Figura 2.25 muestra los resultados de seguimiento del algoritmo expuesto en [Smith05], ante una secuencia obtenida con cámara estática.



Figura 2.25. Resultados obtenidos en la aplicación del algoritmo propuesto en [Smith05]. La estimación de posición arrojada por el filtro para cada persona se marca con un rectángulo de color.

2.2.4. Seguimiento de múltiples objetos con un único estimador multimodal

No existen muchas propuestas dentro de la literatura científica del área que planteen una solución al problema del MTT mediante un sólo estimador aprovechando su multimodalidad. En este punto se analizan las que se han encontrado y presentan más interés en la tesis por su semejanza con la propuesta realizada en esta memoria. Estos trabajos son analizados en mayor profundidad en capítulos posteriores.

El trabajo [Koller-Meier01], desarrollado por un grupo de investigación del ETH de Zurich en conjunto con el GATECH, es uno de los de especial interés en la tesis. En este artículo se describe el “Filtro de Partículas Extendido” (“eXtended Particle Filter”, XPF) que se usa de base del algoritmo de MTT propuesto en esta tesis, como se explica en capítulos posteriores. El

XPF presenta importantes problemas de robustez que obligan a los autores a abandonar su línea de investigación. Debido a ello, en el artículo sólo se muestran resultados de simulación.

Como se explica en el capítulo siguiente, en esta tesis se aporta como solución a la propuesta de Koller-Meier el uso de un clasificador que organice el set de datos de entrada Y_t , con objeto de robustecer el funcionamiento del PF y adecuarlo a las condiciones planteadas. Además, el algoritmo de MTT propuesto en la tesis incorpora también un clasificador de partículas que facilita la obtención de una salida determinística de tracking a partir de la creencia multimodal obtenida con el PF.

El autor del artículo ya descrito [Pérez02] junto a un grupo de investigadores de la Universidad de Cambridge presenta en [Vermaak03] un nuevo sistema de seguimiento de un número múltiple y fijo de objetos k a partir de información de visión, mediante el uso de un solo PF cuyas partículas se clasifican en grupos con el mismo objetivo planteado en la tesis de generar una salida de tracking determinística. El conjunto total de partículas se distribuye entre los objetos, formando lo que los autores denominan “*mixture components*” (“*mezcla de componentes*”) y dando lugar al “*Mixture Particle Filter*” (“*Filtro de Partículas de Mezcla de Componentes*”, MPF), de modo que cada partícula incluye un identificador que la asocia a una u otra componente. La dependencia entre las distintas componentes, que hace que el algoritmo propuesto se diferencie de ser un conjunto de PFs independientes, aparece en el paso de corrección de las partículas.

El algoritmo descrito en [Vermaak03] se pone a prueba con una secuencia de imágenes estáticas de un partido de fútbol, realizándose el seguimiento de un conjunto de jugadores (ver Figura 2.26) localizados inicialmente de forma manual. La etapa de inicialización se utiliza además para extraer el modelo de los objetos a seguir, basado en su histograma de color, que se usa después en un proceso de asociación de tipo NN. El autor no proporciona información sobre el t_{exe} del algoritmo, si bien indica que se utilizaron 200 partículas en los experimentos.



Figura 2.26. Resultados del algoritmo de MTT descrito en [Vermaak03]. Cada jugador objeto del seguimiento se marca en un color diferente en las imágenes con tantos rectángulos como partículas asociadas a él haya en la mezcla de componentes.

El algoritmo de clasificación se usa en este artículo para que el PF pueda ser aplicable al seguimiento varios de objetos, gracias a su carácter multimodal. Esta idea coincide con una de las planteadas en la tesis. Sin embargo en el artículo no se especifica como se podría usar el algoritmo cuando el número de objetos k cambia con el tiempo. En [Vermaak05b] se propone como solución a este problema el uso de una “*probabilidad de existencia*” (“*existence*

probability”) que active el funcionamiento de un KF como estimador del estado de cada objeto implicado en el tracking. En la tesis se propone, sin embargo, clasificar el conjunto de medidas Y_t para, modificando el funcionamiento estándar del PF, incluir el paso de creación y eliminación de objetos a seguir de forma implícita en la ejecución secuencial del estimador probabilístico. En [Vermaak05a] los autores abandonan la idea de usar procesos determinísticos para mejorar la funcionalidad del algoritmo de MTT y abordan el uso del JPDA junto a un set de PFs para implementar varias soluciones de tipo S-JPDAF, como la descrita en [Schulz03b].

En trabajos posteriores del mismo grupo de investigación, como [Ng06], se aborda de nuevo la idea de incorporar un algoritmo de segmentación al estimador bayesiano, en este caso clasificando el conjunto Y_t , y usando la técnica de expansión del vector de estados. El proceso de asociación de las clases a los hipotéticos targets se realiza con un algoritmo JPDA, dando lugar, también en este caso, a un algoritmo de seguimiento de tipo S-JPDAF. La clasificación del set de medidas Y_t se aplica a dos procesos de creación (“*birth*”) y eliminación (“*death*”) de tracks, y por ende, de hipótesis en el filtro bayesiano. Además, se usa un método determinístico basado en la persistencia en el tiempo de las clases de medidas, para iniciar la ejecución de los procesos de creación y eliminación comentados, y así filtrar los outliers, semejante al de validación por ventanas temporales propuesto en [Blackmann99] (ver en el apartado 2.1.2). El proceso específico de creación y eliminación de objetos incluido en los trabajos no permite modificar el número de targets seguidos k en más de uno por iteración, por lo que se imposibilita el funcionamiento del algoritmo en entornos muy dinámicos y poblados.

Al margen de [Vermaak03], ninguna de las publicaciones presentadas hasta el momento por este grupo de investigación muestra resultados reales mejores que los mostrados en este primer artículo, lo que imposibilita la comparativa del resto de trabajos referidos con la propuesta presentada en esta tesis.

Por otro lado, un grupo de investigadores de la Universidad British Columbia presentan el que es uno de los trabajos más interesantes en el contexto de la tesis. En [Okuma04] los autores proponen el uso de un clasificador de medidas para usar un MPF ([Vermaak03]) en una aplicación MTT con número variables de objetos k . El clasificador propuesto (detector “*Adaboost*” basado en el aprendizaje de modelos de color, referido también en [Kobayashi06]) se utiliza además para detectar los objetos en la escena y como base del proceso de asociación necesario para el desarrollo del seguidor.

El algoritmo resultante es llamado por los autores “*Boosted Particle Filter*” (BPF), y según los propios autores es una mejora del expuesto en [Vermaak03]. Este algoritmo es el encontrado en la literatura del área que presenta una mayor similitud con el propuesto en la tesis, por lo que su funcionalidad se analiza en detalle más adelante en la tesis.

En [Okuma04], la funcionalidad del BPF se comprueba en el seguimiento de un conjunto variable de jugadores de jockey sobre hielo en distintas escenas de un partido (ver Figura 2.27). El detector “*Adaboost*” en se entrena para aprender las características diferenciales de color de los elementos de la escena: blanco para el fondo y el color de las camisetas y pantalones de los

jugadores para estos últimos. El modelo de observación es, por tanto, demasiado específico para alcanzar las especificaciones planteadas en esta tesis.

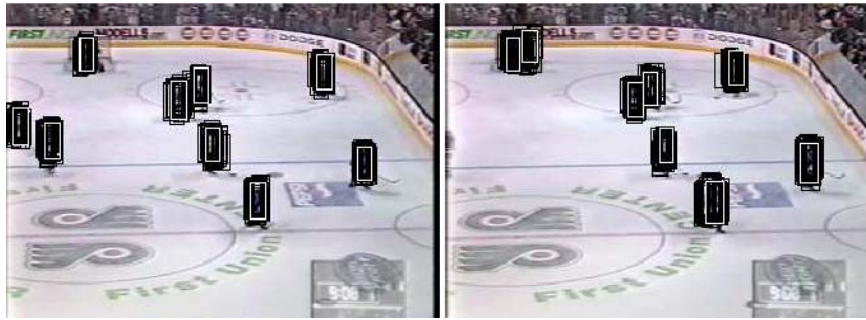


Figura 2.27. Resultados del algoritmo de MTT propuesto en [Okuma04]. Los rectángulos blancos representan a la salida del algoritmo “Adaboost”, mientras que los negros representan el valor de las partículas. En estas dos imágenes se aprecia la capacidad del algoritmo de reaccionar correctamente ante la desaparición de objetos.

2.3. Recapitulación y conclusiones

En este capítulo se realiza una revisión de trabajos relacionados con la temática de investigación de la tesis. La revisión incluye un resumen teórico de las técnicas de estimación y asociación (probabilísticas y determinísticas) más usadas en la solución del problema de MTT.

A partir de este resumen, la revisión de trabajos permite tener una visión completa de los algoritmos básicos y las mejoras que sobre ellos han sido propuestas por la comunidad científica para obtener un algoritmo de seguimiento de múltiples objetos como el propuesto en la tesis.

Por otro lado, el análisis teórico permite orientar la revisión de trabajos a aquellos que se basan en el uso del PF como estimador. Esta elección se debe a su facilidad de implementación, incluso para sistemas de modelo complejo (hecho que no se da en la aplicación de interés), a la capacidad que tiene de manejar múltiples hipótesis de seguimiento bien sea representando a todos o a cada uno de los objetos implicados en la tarea de MTT, y a su robustez en la estimación bajo situaciones complejas de ruido y de densidad variable de medidas. Tanto es así, que en los últimos diez años el PF ha sido el centro de atención de la mayor parte de propuestas realizadas por la comunidad científica para la resolución de la tarea de MTT ([Ng06]).

En la revisión de cada trabajo se incluye, además del análisis de funcionalidad en el que se especifica la base teórica de la propuesta, el o los puntos en los que el trabajo aporta una novedad a la solución de MTT perseguida. Además se incluye por cada trabajo un resumen de resultados, en el que se analiza si se alcanzan las especificaciones de flexibilidad (sobre todo en cuanto al sensor y el modelo de observación usado), fiabilidad del algoritmo en entornos complejos típicos de la robótica personal, robustez (frente a la incertidumbre del estado \vec{v}_t y el ruido y distribución desigual de las medidas \vec{o}_t) y tiempo real planteadas en el capítulo anterior.

En la revisión se hace, además, especial hincapié en analizar las propuestas que usan el carácter multimodal del PF y la combinación de métodos probabilísticos y determinísticos para la estimación de la posición de un número variable de objetos k , pues estas son las

aportaciones principales de la tesis. En esta línea, en capítulos posteriores se analiza en detalle trabajos como [Koller-Meier01], [Vermaak03], [Okuma04] o [Ng06].

Tal y como se expone en este capítulo, se pueden diferenciar dos tendencias fundamentales entre las propuestas más importantes de la comunidad científica para resolver el problema de MTT: el uso de seguidores individuales que se relacionan en el proceso de asociación y el uso de un único estimador de vector de estado aumentado con el de cada uno de los objetos a seguir.

Ambas soluciones presentan como mayor inconveniente un incremento de la carga computacional del estimador que crece de forma proporcional e incluso exponencial con el número de objetos a seguir k , y con ella su t_{exe} , lo que imposibilita que se ejecuten en tiempo real para más de cuatro o cinco objetos. Es por ello, que el principal objetivo de los distintos trabajos desarrollados por la comunidad científica en esta área de investigación se centra en diseñar versiones del PF básico que minimicen la carga computacional, surgiendo así por ejemplo el RBPF que se usa en [Torma03], [Schulz03a], [Khan04] o [Sarkka07]. Como se demuestra en capítulos posteriores, el algoritmo propuesto en esta tesis tiene un t_{exe} constante e independiente del número de objetos implicados en el seguimiento.

Además, a pesar de las continuas referencias al uso del carácter multimodal del PF en el modelado de múltiples targets con una única PDF, a lo largo de los últimos años (p. ej. en [Hue02]), la extensión de un único PF para su uso en el seguimiento de un número variable de objetos k , tal y como se propone en esta tesis, no ha sido resuelta con éxito aún.

Otro de los problemas que se mencionan repetidamente en los trabajos analizados en este capítulo, es la falta de robustez del PF en su aplicación a la tarea de seguimiento en situaciones complejas por el ruido o la cantidad de elementos de distracción o clutter que aparecen en la escena de la que se extraen las medidas de posición. Para resolver este problema, que se analiza en mayor profundidad en el capítulo 4, se modifica el PF básico dando lugar a algoritmos como el “*ICondensation*” ([Isard98b]), el principio de exclusión de [MacCormick99] o el PF basado en mezcla de gaussianas ([Vermaak03]).

En esta línea, una de las aportaciones más importantes de la presente tesis es la demostración de que la robustez del PF en la tarea de estimación multimodal se puede incrementar combinando su funcionalidad con la de algoritmos determinísticos que clasifican el set de medidas Y_t a la entrada del estimador, como se explica anteriormente. De este modo, se posibilita una ejecución más rápida y a la vez más robusta de los seguidores basados en PF.

Otra forma de resolver el problema de robustez mencionado es el de usar un modelo complejo de observación, dando lugar a algoritmos como “*BraMBle*” de [Isard01], el “*Subordinated Condensation*” de [Tweed02] o el “*MRF-augmented PF*” de [Khan05] que resuelven además el problema de asociación implícito en la aplicación de MTT.

Tal y como ya se ha explicado, en esta tesis se propone el uso de modelos sencillos de observación, que si bien complican el problema de estimación, hacen el algoritmo más rápido y

más flexible para aplicaciones de seguimiento múltiple, basadas en medidas de posición de tipología distinta a la de visión, usada en esta tesis.

En general, como se concluye de los trabajos analizados, las soluciones de seguimiento se centran en el diseño de sistemas de detección/observación cada vez más complejos, sobre todo en el área del tracking basado en sistemas de visión, simplificando el algoritmo de estimación y asociación usados para la predicción del movimiento de los objetos. Este tipo de propuestas son efectivas y rápidas en escenas simples, pero la falta de un proceso de filtrado más elaborado avoca a la complicación de los procesos de detección, con objeto de que los algoritmos resultantes sean fiables en escenas complejas (ruidosas, muy pobladas y con múltiples oclusiones), y esto conlleva la pérdida de ejecución en tiempo real para un número elevado de objetos a seguir k , en la mayor parte de los casos. Además, la adaptación del algoritmo de MTT a un número variable de objetos se consigue generalmente mediante procesos añadidos de unión o separación de las distintas hipótesis del vector de estado definido en la estimación.

En esta tesis se defiende la idea de que el uso de algoritmos de estimación y asociación más elaborados, permite conseguir la fiabilidad deseada manteniendo la especificación de tiempo real, y añadiendo a la solución final otras características como la robustez y la flexibilidad, muy interesantes en el área de la robótica personal.

A la vista de la revisión de trabajos anteriores incluida en este capítulo, se puede concluir que la tarea que se plantea resolver en la tesis no está resuelta de forma completa, y que las aportaciones que en ésta se proponen mejoran las especificaciones de funcionalidad de la aplicación de interés.

CAPÍTULO 3

DESCRIPCIÓN GENERAL DEL SISTEMA

Realizado el análisis del estado del arte en la temática de interés, en este capítulo se expone una descripción general del sistema de seguimiento de múltiples objetos propuesto.

En los capítulos 4 y 5 se analizan en detalle cada una de las partes que componen el proceso de seguimiento planteado en la tesis, de modo que la introducción que aquí se presenta permite tener una visión global de la esta propuesta.

3.1. El “*Filtro de Partículas Extendido con Proceso de Clasificación*” (XPFCP)

Tal y como se explica en capítulos anteriores, la propuesta de seguimiento de múltiples objetos presentada se basa en la combinación de métodos probabilísticos de estimación con métodos determinísticos de asociación que además robustecen el funcionamiento del estimador multimodal. El algoritmo de MTT resultante permite obtener y seguir la posición de un número variable de objetos de forma fiable incluso en situaciones complejas (entornos muy poblados y ruidosos), a partir de un conjunto más o menos denso de medidas de posición Y_t de los objetos a seguir, que en este caso se obtienen con un sistema de visión, si bien el algoritmo es suficientemente flexible como para adaptarse a modelos de actuación y observación distintos (más complejos e incluso no lineales) que los usados en la tesis.

La Figura 3.1 muestra un diagrama general de funcionamiento del sistema de seguimiento de múltiples objetos propuesto.

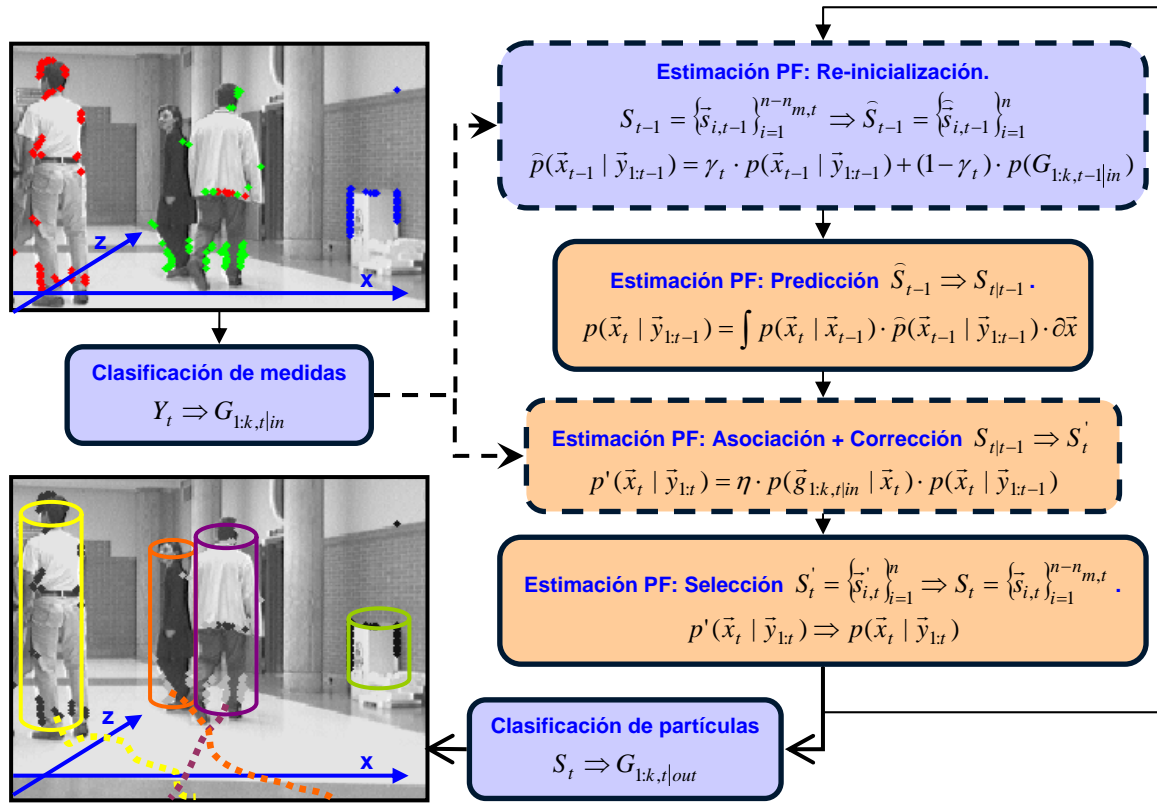


Figura 3.1. Diagrama de bloques general del XPFCP propuesto en la tesis. Los bloques en color naranja representan los pasos básicos del PF y los de color azul los nuevos que se incorporan a éste para dar lugar al algoritmo propuesto. Con línea discontinua se marcan los bloques en los que se usa la información procedente de la clasificación de medidas $Y_t \Rightarrow G_{1:k,t|in}$ o de partículas $S_t \Rightarrow G_{1:k,t|out}$. Se incluye una imagen obtenida en un experimento real, sobre la que se representa con puntos (coloreados en función de su coordenada x) el set Y_t y la salida determinística del algoritmo caracterizada por un cilindro de color diferente para cada objeto, cuya posición y velocidad es obtenida por el XPFCP en cada instante y conforma su trayectoria en el espacio y el tiempo, con línea punteada y en el mismo color.

En la Figura 3.1 se observa que el algoritmo probabilístico utilizado como base es un PF (con sus tres pasos básicos de predicción, corrección y selección¹), cuyo carácter multimodal se usa para modelar con una única PDF la posición del número variable de objetos que se siguen en cada instante ² $k_{out,t}$. Para que la creencia $p(\bar{x}_t | \bar{y}_{1:t})$ que obtiene el PF permita caracterizar al número variable de targets $k_{out,t}$ implicados en la tarea de MTT, es necesario introducir un paso nuevo en el PF, denominado de “re-inicialización” (en azul en la figura).

¹ El paso de selección se añade en la versión básica del PF respecto a la del filtro de Bayes básico, explicada en el capítulo 2, para evitar un problema de degeneración del set de partículas. Este hecho se analiza en el capítulo 4.

² A partir de este punto de la tesis el número de objetos seguidos en cada momento se identificará como $k_{out,t}$ por obtenerse de la salida del clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$, y diferenciarse de $k_{in,t}$ que se usará para identificar el número de clases en que se organiza el set $Y_t \Rightarrow G_{1:k,t|in}$ de entrada al seguidor.

El PF así modificado (también denominado “*Filtro de Partículas Extendido*” o “*eXtended Particle Filter*”, XPF) fue introducido en el trabajo [Koller-Meier01], como se explica en el capítulo 2. En el capítulo siguiente se demuestra que el XPF es un algoritmo poco robusto, pues pierde las hipótesis de estimación fácilmente ([Schulz03b]). Para resolver este problema, y otros relacionados con la fiabilidad del PF, la comunidad científica propone distintas alternativas que son analizadas igualmente en el capítulo 4.

En esta tesis se propone tomar como base el XPF y añadirle un proceso que modifique su funcionamiento y permita que éste se adapte de forma robusta a la tarea de estimación multimodal. Este proceso determinístico, que se combina con el probabilístico anterior, es un clasificador del set de medidas de posición Y_t que se obtienen en cada momento del conjunto de objetos a seguir (en azul en la Figura 3.1). La información extraída del segmentador de medidas $G_{j,t|in} / j=1:k_{in,t}$ se usa en la etapa de corrección y en el nuevo paso de re-inicialización, como se muestra con línea discontinua en la figura, permitiendo robustecer el sistema de seguimiento multi-objeto, y reforzando la tarea de asociación incluida en él.

El algoritmo así resultante, cuya funcionalidad muestra de forma general la Figura 3.1, es nombrado en la tesis como “*Filtro de Partículas Extendido con Proceso de Clasificación*” o “*eXtended Particle Filter with Clustering Process*”, XPFCP.

En los capítulos siguientes se justifica teórica y empíricamente que el XPFCP resuelve el problema de MTT planteado, cumpliendo todas las especificaciones planteadas al principio de la tesis. Además en el capítulo 7 se comparan cuantitativa y cualitativamente los resultados arrojados por la solución propuesta con los generados por las alternativas (el JPDAF y el S-JPDAF) que, según se desprende del análisis del estado del arte, son más comúnmente usadas por la comunidad científica en la aplicación de interés.

En la Figura 3.1 se muestra, junto al flujograma general del XPFCP, una escena de ejemplo del proceso de MTT, extraída de un experimento real, en la que aparecen cuatro objetos (tres personas y una papelera).

Sobre la imagen superior de la Figura 3.1 se muestra el conjunto Y_t de medidas de posición (representados por puntos cuyo color indica el valor umbralizado de su coordenada x ; rojo si la medida está a la izquierda en la escena, verde si está en el centro y azul si está a la derecha) a partir del que se ejecuta el XPFCP. Tal y como se observa, la matriz de datos está formada por un set de puntos desigualmente repartido entre los objetos, con outliers (como el que aparece sobre la papelera en la pared), y con errores de medida (como los puntos rojos sobre la persona de la derecha, que tienen, evidentemente, un valor de x incorrecto).

Sobre la imagen inferior de la Figura 3.1 se representa el resultado de seguimiento generado por el XPFCP. Éste incluye un rastro de posición, en el espacio XZ , del camino recorrido por cada uno de los objetos encontrados en el entorno (identificado con un color diferente por cada objeto) desde el principio del experimento hasta el instante de captura de la imagen, y su posición y velocidad actual caracterizada por un cilindro.

La salida determinística del XPFCP se obtiene organizando en clases el conjunto de n partículas ($S_t = \{\bar{s}_{i,t} / i = 1:n\}$) que caracteriza a la creencia $p(\bar{x}_t | \bar{y}_{1:t})$, para lo cual es necesario aplicar un nuevo proceso de clasificación, $S_t \Rightarrow G_{1:k_t | out}$. Este proceso, que nada tiene que ver con el descrito en párrafos anteriores, segmenta la PDF de salida en función de sus picos de máxima probabilidad, que representan el estado \bar{x}_t del número variable de objetos $k_{out,t}$ seguidos por el algoritmo de MTT propuesto. Esto es así teniendo en cuenta que, tal y como se explica en capítulos anteriores, en la tesis se propone usar el carácter multimodal del PF para caracterizar hipótesis muy probables del vector de estado \bar{x}_t de los diferentes objetos a seguir.

En la Figura 3.2 se aprecia el carácter multimodal de la creencia $p(\bar{x}_t | \bar{y}_{1:t})$, a través del conjunto de partículas S_t que se obtiene a la salida del XPFCP en el instante del experimento real mostrado en la Figura 3.1. En la imagen izquierda de ésta se muestra el valor de las componentes de posición del set de partículas S_t en el plano XZ en función de sus pesos \bar{w}_t . Los cuatro picos que aparecen en la PDF discretizada $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$ permiten caracterizar a los cuatro objetos en la escena de la Figura 3.1. En la imagen derecha se muestra la proyección de estas componentes en el mismo plano XZ . En esa imagen no es posible distinguir los cuatro modos de $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$.

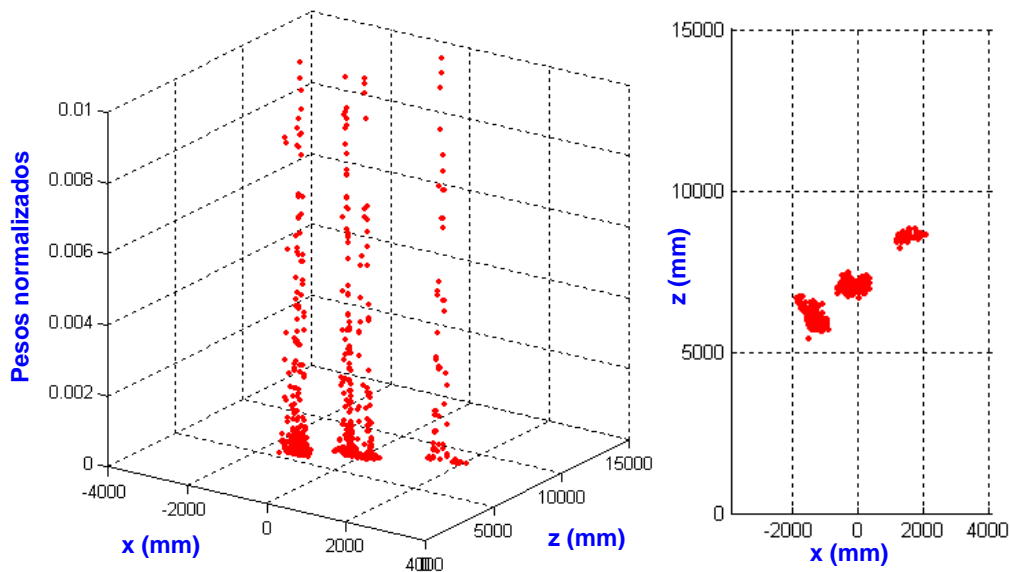


Figura 3.2. Representación 3D (a la izquierda) y 2D (a la derecha) del valor de las componentes x y z del conjunto de partículas S_t y sus correspondientes pesos \bar{w}_t a la salida del XPFCP, en el instante del experimento real mostrado en la Figura 3.1.

El uso de un algoritmo que permita obtener una salida determinística a partir de $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$ es una técnica muy recurrida por los usuarios del PF. La razón es que la información que proporciona el set S_t a la salida del filtro no permite sacar fácilmente una conclusión sobre el proceso de estimación, como se aprecia en la Figura 3.2.

En las aplicaciones en las que el PF se usa para modelar una PDF multi-hipótesis sobre un único vector de estado (ya sea éste simple \bar{x}_t y caracterice a un único objeto, o extendido χ_t y caracterice a todos los implicados en la tarea de MTT), el método para obtener una salida determinística suele consistir en calcular algún estadístico de la densidad discreta $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$, como la media o el máximo (convirtiéndose entonces en un estimador tipo MAP, “*Maximum A Posteriori*”). Sin embargo, en la propuesta planteada en esta tesis, la PDF representada por el set S_t caracteriza a un número múltiple y variable de vectores de estado, tal y como se aprecia en la Figura 3.2, por lo que es necesario usar un algoritmo de clasificación de las partículas para obtener la salida determinística buscada a través de las clases generadas por éste $S_t \Rightarrow G_{1:k,t|out}$. Esta salida se representa con cilindros sobre los objetos seguidos en la Figura 3.1.

Tal y como se analiza en el capítulo 2, existen varios trabajos en la literatura del área en los que se propone la combinación de PFs y algoritmos de clasificación para resolver el problema de MTT de interés. Sin embargo, ninguna de estas propuestas plantea el uso de dos procesos de segmentación en los puntos y con los objetivos planteados en la tesis. En capítulos posteriores se estudian de forma específica trabajos analizados anteriormente, como [Okuma04] o [Ng06] en los que se plantea el uso de un clasificador del set de medidas $Y_t \Rightarrow G_{1:k,t|in}$, con una funcionalidad semejante a la propuesta en la tesis; o como [Vermaak03] donde se describe el uso de un segmentador de partículas $S_t \Rightarrow G_{1:k,t|out}$ a la salida del PF, de forma también similar a la aquí propuesta.

En el capítulo 5 se presentan varios algoritmos diseñados para desempeñar las dos tareas de clasificación implicadas en las etapas del XPFCP. En dicho capítulo se analiza la robustez de estos segmentadores y su idoneidad de uso como clasificador de medidas $Y_t \Rightarrow G_{1:k,t|in}$, a la entrada del estimador multimodal, o de partículas $S_t \Rightarrow G_{1:k,t|out}$ a la salida del mismo.

3.2. Descripción general de los algoritmos que conforman el XPFCP

A la vista del diagrama general de funcionamiento del XPFCP (Figura 3.1), en este punto se analizan los dos algoritmos fundamentales que lo conforman, de modo que esta descripción sirva de introducción de los conceptos que se estudian en los capítulos siguientes.

3.2.1. Contenido probabilístico de la propuesta. El filtro de partículas como solución óptima de estimación al problema de seguimiento

El PF es un estimador probabilístico que basa su funcionamiento en realizar una representación estocástica discreta por medio de un conjunto S_t de n muestras ($S_t = \{\bar{s}_{i,t} / i = 1:n\}$) del estado del sistema a estimar \bar{x}_t . Las muestras están además ponderadas por una probabilidad de certidumbre $\bar{s}_{i,t} = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$, conformando así una PDF discreta que caracteriza a la creencia que se desea estimar $S_t \cong p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$. En el capítulo 4 se incluye un análisis exhaustivo de funcionamiento del PF, ya introducido en el capítulo 2.

Tal y como ya se explica en párrafos anteriores, es necesario modificar la funcionalidad del PF básico con objeto de adaptarlo de modo que la densidad multimodal representada por el conjunto de partículas permita caracterizar de forma robusta el estado del número variable de objetos $k_{out,t}$ implicado en el proceso de MTT en cada instante.

En los capítulos 4 y 6 se incluye también un estudio teórico detallado sobre estas modificaciones necesarias, que se describen de forma resumida a continuación para comprender el funcionamiento de la propuesta:

1. *Paso de re-inicialización*: Esta es la etapa que se añade al PF básico para que sea posible la incorporación de nuevas hipótesis de seguimiento en la creencia discretizada que genera como la salida. Tal y como se aprecia en la Figura 3.1, la funcionalidad de este paso consiste en añadir al set $S_{t-1} \equiv p(\bar{x}_{t-1} | \bar{y}_{1:t-1})$ que caracteriza a la densidad a posteriori obtenida en el instante temporal anterior $t-1$, un conjunto nuevo de partículas, a partir de la clasificación de las medidas de ese mismo instante ($Y_{t-1} \Rightarrow G_{1:k,t-1|in}$), generándose así un nuevo set \hat{S}_{t-1} .

Para que el conjunto resultante \hat{S}_{t-1} tenga un número constante de partículas n , es necesario modificar el paso de selección del algoritmo, tal y como se explica más abajo, con la idea de que el set S_{t-1} llegue a este paso formado por $n - n_{m,t-1}$ partículas, y en él se le añadan $n_{m,t-1}$. El parámetro γ_t permite relacionar las cantidades $n_{m,t-1}$ y n , de modo que

$$\gamma_t = \frac{n_{m,t-1}}{n}.$$

Esta modificación conlleva que la PDF $\hat{p}(\bar{x}_{t-1} | \bar{y}_{1:t-1})$ a partir de la que se desarrolla las etapas básicas del PF quede como se muestra en la expresión <3.1>, ya expuesta en la Figura 3.1. Los efectos que tiene el uso de esta densidad en el filtro bayesiano, así como el método específico que se usa para aumentar el número de partículas en S_{t-1} , son cuestiones analizadas en detalle en los capítulos siguientes.

$$\hat{p}(\bar{x}_{t-1} | \bar{y}_{1:t-1}) = \gamma_t \cdot p(\bar{x}_{t-1} | \bar{y}_{1:t-1}) + (1 - \gamma_t) \cdot p(G_{1:k,t-1|in}) \quad <3.1>$$

2. *Paso de selección*: Esta etapa se ejecuta una vez se han desarrollado los pasos de predicción y corrección del estimador, y se cuenta con un conjunto de muestras ponderadas de la creencia $S'_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$. En este paso se seleccionan $n - n_{m,t}$ partículas en función de su peso, de modo que el de re-inicialización antes descrito pueda ejecutarse sobre el conjunto resultante $S_t = \left\{ \bar{x}_t^{(i)}, \frac{1}{(n - n_{m,t})} \right\}_{i=1}^{n - n_{m,t}}$ en el instante temporal siguiente $t + 1$.

La necesidad de incluir un paso de selección ya en la versión básica del PF está ampliamente discutida en la bibliografía relacionada con el filtro bayesiano, y las conclusiones alcanzadas se resumen en el capítulo 4.

Una vez analizadas las modificaciones más importantes que se realizan en el PF básico, para alcanzar las especificaciones requeridas en la tesis, parece conveniente introducir aquí un

listado de *razones por las que se elige el PF como núcleo del algoritmo propuesto*, antes de proceder en el capítulo 4 con un análisis exhaustivo sobre éste. Este listado sirve de resumen de los razonamientos expuestos en capítulos anteriores:

- *Multimodalidad.* El estimador basado en un PF permite la incorporación de nuevos modelos en el proceso de estimación con facilidad sin que esto tenga una repercusión negativa en lo que se refiere a la fiabilidad, exactitud o eficiencia de la estimación. De esta forma la estimación de la posición es realizada con un único algoritmo y en tiempo constante, independientemente del número de sistemas implicados. Para ello es necesario incorporar un método de asociación suficientemente robusto, tal y como se comenta más adelante en este punto. La razón más importante de uso del PF como estimador en la tarea de MTT propuesta es, por tanto, su capacidad para estimar PDFs multimodales, como la mostrada en la Figura 3.2, necesarias para la implementación del seguidor en un único filtro, con un coste computacional bajo ([Gordon97]).
- *Flexibilidad.* En la mayor parte de aplicaciones, al contrario de lo que sucede en la de interés, la razón principal de uso de un PF como estimador es el carácter no lineal del modelo que se desea estimar. En esta tesis, esta razón no es la principal, como se comenta en el párrafo anterior, pues además el modelo usado en la aplicación de interés es lineal. El uso de un PF permite, sin embargo, utilizar el algoritmo de tracking propuesto en aplicaciones en las que el modelo sea no lineal, sin detrimento de la fiabilidad o exactitud del seguidor.
- *Fiabilidad.* Finalmente, el carácter probabilístico del PF es la otra razón fundamental por la que se plantea usar un filtro de este tipo para la tarea de estimación implicada en la aplicación de MTT. Este carácter estocástico facilita que esta tarea se realice de forma fiable ante hipótesis muy débiles, como las procedentes de un sistema de medida como el planteado, y con una alta tolerancia a los fallos. De hecho las soluciones basadas en el PF han mostrado mayor fiabilidad ante problemas de estimación que la de soluciones basadas en el KF y sus versiones de aplicación no lineal EKF y UKF ([Hue02] o [Schulz03b]).

La posibilidad de aprovechar el carácter multimodal del PF, base de la propuesta de MTT planteada en la tesis, se expone en varios trabajos anteriores como [Isard98a], artículo revisado en el capítulo 2, caracterizado además por ser el primero en el que se propone aplicar un PF (denominado allí algoritmo “*Condensation*”) en tareas de seguimiento en imágenes.

La Figura 3.3, extraída de [Isard98a], muestra como la creencia multimodal $p(u_t | \bar{y}_{1:t})$ que genera el estimador (solo se muestra su valor en el eje horizontal, u de la imagen) se adapta para modelar la posición $\bar{x} = u$ de las tres personas que aparecen en la escena (aparecen tres picos en $p(u_t | \bar{y}_{1:t})$ que se mueven de acuerdo a como lo hacen las personas en el eje u de la imagen). Sin embargo, tal y como se observa en la figura, la PDF multimodal $p(u_t | \bar{y}_{1:t})$ no es muy robusta, pues tiende a afilarse para valores medios de u en el espacio representado, debido a que los objetos en su movimiento pasan con mayor asiduidad por esa zona y, por tanto, la verosimilitud $p(\bar{y}_t | u_t)$ incrementa en mayor medida ese pico de probabilidad de $p(u_t | \bar{y}_{1:t})$ que el resto, con lo que las otras hipótesis de tracking se ven empobrecidas.

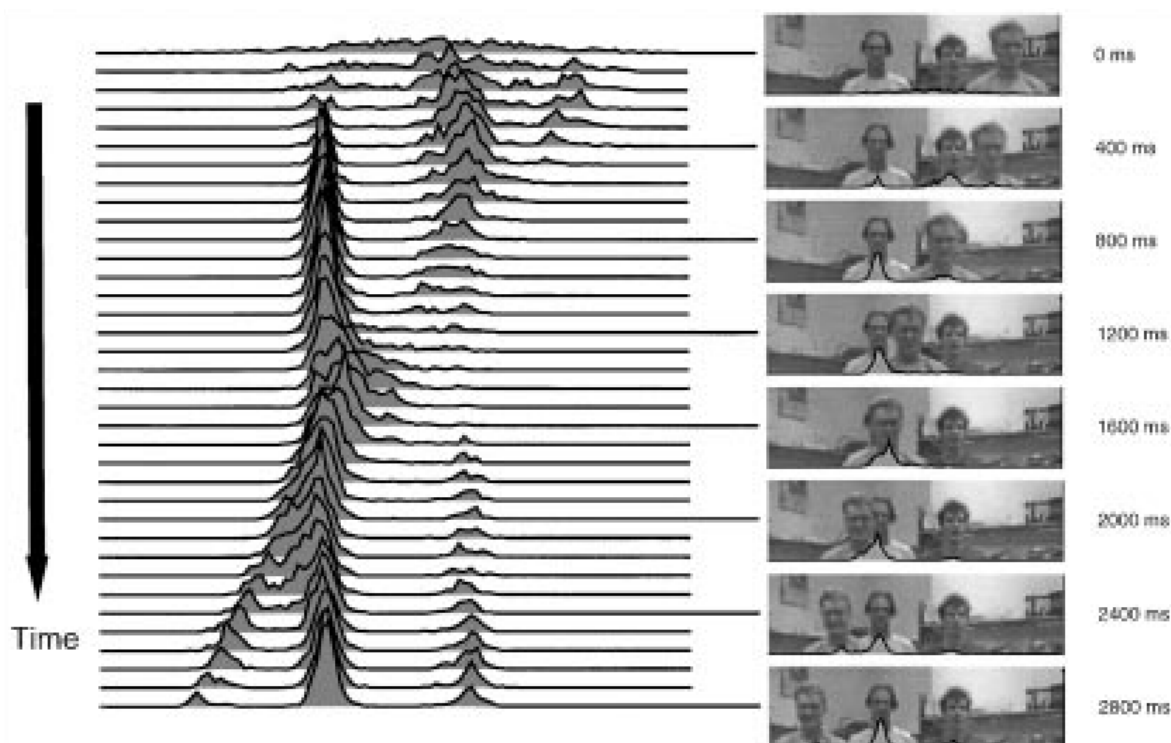


Figura 3.3. Multimodalidad de la creencia $p(\bar{x}_t | \bar{y}_{1:t})$ generada por el PF en una tarea de seguimiento de múltiples objetos, extraída de [Isard98a]. A la izquierda de la imagen se muestra la evolución temporal de la coordenada u de la PDF $p(u_t | \bar{y}_{1:t})$ que modela la posición $\bar{x} = u$ de cada una de las personas en la imagen, que se expone a su vez a la derecha para una de cada cinco ejecuciones del filtro. El instante temporal de proceso de cada imagen se muestra al lado de estas.

Existen métodos muy diversos propuestos por la comunidad científica para intentar resolver este problema de “empobrecimiento de la creencia” $p(u_t | \bar{y}_{1:t})$, mostrado en la Figura 3.3, que van desde el aumento del número de partículas n (con objeto de que siempre haya alguna asociada a cada hipótesis), lo que como contrapartida incrementa el tiempo de ejecución t_{exe} del estimador, hasta los algoritmos “*ICondensation*” y “*PF con muestreo segmentado*”, comentados en el capítulo 2. En el capítulo 4 se analizan estas y otras propuestas para disminuir el problema de empobrecimiento de la multimodalidad del PF.

El hecho es que el problema de empobrecimiento descrito se usa en varios de los trabajos analizados en el capítulo anterior (como en [Schulz03b]) para justificar el empleo de un algoritmo de seguimiento para cada objeto, y para concluir que el PF así usado no puede utilizarse para modelar el estado de un número variable de objetos $k_{out,t}$, debido a su baja robustez. Para resolver este problema, la propuesta de MTT presentada en esta tesis y mostrada de forma general en la Figura 3.1 incluye el proceso determinístico de clasificación de medidas $Y_t \Rightarrow G_{1:k,t|in}$ que elimina el problema de empobrecimiento comentado tal y como se expone en el apartado siguiente y se demuestra a lo largo de la tesis.

En [Hue02] se comenta también que el modo de usar el PF planteado en la tesis para la tarea de MTT solo es válido, evidentemente, en el caso en que el modelo de actuación $p(\bar{x}_t | \bar{x}_{t-1})$ de

todos los objetos sea el mismo. Esta restricción no supone ningún problema en la aplicación de interés, al usar el modelo de velocidad constante (CV) expuesto en el capítulo 2, y es, además una especificación usada en la mayor parte de trabajos del área.

Finalmente, en distintas referencias básicas de la literatura del área, como [MacCormick99] [Khan04] o [Ng06], se explica que el mayor escollo en el uso de un PF para tareas de estimación es su t_{exe} , que depende del número de partículas n del mismo. Este tiempo crece exponencialmente con la dimensión de \bar{x} ([Dellaert03] o [Schulz03b]), y por ende, con el número de objetos implicados en el proceso de seguimiento, en el caso de usar un algoritmo basado en la extensión del vector de estado, en el que el vector a estimar es $\chi_t = \bigcup_{j=1}^{k_{out,t}} \bar{x}_{j,t}$, y evidentemente, de forma lineal con el número de objetos $k_{out,t}$ si el algoritmo de tracking se basa en el uso de un estimador por objeto.

Con la propuesta de tracking hecha en esta tesis, se minimiza el número de partículas n necesarias en el PF para conseguir resultados suficientemente fiables, pues se propone usar un solo estimador con vector de estado simple ($\bar{x}_t = [x_t \ y_t \ z_t \ vx_t \ vz_t]^T$ con la posición de un solo objeto), independientemente del número de objetos implicados en el proceso. El problema de carga computacional asociada al uso de PFs para la aplicación de MTT queda así eliminado, manteniéndose además un t_{exe} constante, independientemente de $k_{out,t}$. En el capítulo 6 se incluye un análisis del número de partículas n necesarias y el consiguiente t_{exe} del XPFCP en función del número de objetos implicados en la tarea de seguimiento.

3.2.2. Contenido determinístico de la propuesta. El clasificador como solución óptima de asociación y robustecimiento del algoritmo de seguimiento

En esta tesis se propone usar un algoritmo de clasificación para organizar el conjunto de medidas de entrada del sistema de MTT $Y_t \Rightarrow G_{1:k,t|in}$, y emplear el resultado de clasificación en el robustecimiento de la creencia multimodal generada por el XPFCP, y en el proceso de asociación de tipo NN que se incluye así de forma implícita en el estimador.

El uso de un algoritmo determinístico como proceso de asociación se justifica en el hecho, comentado en el capítulo 1, de que la combinación del carácter probabilístico del estimador de posición y el determinístico del método de asociación elegido permite alcanzar la flexibilidad, robustez y fiabilidad deseada para la propuesta de MTT, manteniendo la carga computacional del seguidor multi-objeto suficientemente baja como para poder ejecutarlo en tiempo real para un número alto de targets.

Además, teniendo en cuenta la necesidad de usar un proceso de asociación, que tal y como se describe en el capítulo 2, viene impuesta por la aplicación de MTT, conviene organizar el conjunto de medidas de entrada al algoritmo, para mejorar la fiabilidad de la asociación disminuir su coste computacional y facilitar así la consecución de las especificaciones establecidas en la tesis.

Estos razonamientos se confirman en trabajos anteriores (analizados en el capítulo 2) y se completan con los que se exponen a continuación, para *justificar el uso de un algoritmo de clasificación determinístico como base del proceso de asociación necesario en la aplicación de MTT perseguida*:

- *Robustecimiento del PF multimodal.* Tal y como se observa en la Figura 3.3, el uso de un único PF para estimar el estado \bar{x}_t de un número variable de modelos $k_{out,t}$, a través de la multimodalidad de la creencia obtenida con él $p(\bar{x}_t | \bar{y}_{1:t})$, constituye una solución de MTT poco robusta, si no se incluye ningún módulo más en el sistema de tracking. La modificación del PF básico propuesta en [Koller-Meier01] permite añadir y eliminar hipótesis que caractericen el estado \bar{x}_t de los objetos según aparecen y desaparecen del entorno. Sin embargo, el algoritmo así obtenido sigue siendo poco robusto y sufriendo el problema de empobrecimiento comentado en puntos anteriores. La información obtenida del proceso de clasificación de medidas $G_{j,t|in} / j=1:k_{in,t}$ se usa en la propuesta presentada en esta tesis para modificar de forma controlada la probabilidad de las hipótesis modeladas por la creencia $p(\bar{x}_t | \bar{y}_{1:t})$, robusteciendo así la funcionalidad del sistema de MTT propuesto.

El algoritmo de segmentación actúa además de filtro de la PDF de verosimilitud $p(\bar{y}_t | \bar{x}_t)$ que se obtiene del modelo de observación, eliminando de ésta los valores que posiblemente estén relacionados con ruidos de medida \bar{o}_t , y aumentando la certeza de aquellas hipótesis sensadas pobremente por error, debido a diversas condiciones del entorno (como oclusiones parciales o totales). Esta es una de las razones fundamentales de uso de un proceso determinístico de asociación en la propuesta de MTT realizada en la tesis.

- *Novedad del carácter determinístico.* Teniendo en cuenta que la aplicación de métodos determinísticos al problema de asociación está menos extendida entre la comunidad científica, tal y como se extrae del análisis expuesto en el capítulo 2, y que éstos aportan una serie de ventajas importantes como las expuestas en párrafos anteriores, el diseño de un algoritmo de MTT basado en métodos determinísticos de asociación, constituye una interesante aportación científica de la tesis.
- *Idoneidad en tracking sobre imágenes.* Tal y como se explica en el capítulo 1, el uso de sensores de visión en aplicaciones de MTT es, en la actualidad, una línea de gran interés en el área de la robótica personal, razón que lleva a usar este tipo de información en los desarrollos mostrados en la tesis. La alta densidad de datos generada por estos sensores juega un papel positivo en el uso de procesos probabilísticos, típicos en aplicaciones de MTT por su eficiente manejo de la incertidumbre.

Esta afirmación se justifica en el hecho de que el filtrado probabilístico mejora su robustez cuanto mayor es la base de información procedente del entorno. Sin embargo, para alcanzar la especificación de tiempo real en esta situación es necesario introducir una tarea de segmentación de los datos de visión. Esta clasificación permite discriminar los datos que aportan información de interés en el entorno, eliminando aquellos que no lo hacen (por ser

redundante u outlier) y que, sin embargo, incrementan el t_{exe} de la aplicación. El uso de algoritmos de clasificación es, por tanto, especialmente relevante en aplicaciones de MTT basadas en sensores de visión.

- *Simplificación de las tareas y modelos implicados en el seguimiento.* En la revisión de trabajos realizada en el capítulo 2 se describen diferentes algoritmos de segmentación de la información contenida de la imagen, propuestos por la comunidad científica para la aplicación de seguimiento ([Okuma04], [Branson05] o [Chang06], entre muchos otros). Sin embargo, tal y como allí se comenta, estos procesos son complejos y muy específicos en función de la tipología del objeto a seguir, lo que restringe la flexibilidad de la aplicación de MTT y puede impedir su ejecución en tiempo real para más de cuatro o cinco objetos.

La carga computacional del seguidor se disminuye simplificando los modelos de actuación y observación, el algoritmo de asociación, o el de estimación para múltiples sistemas. En la propuesta presentada en esta tesis se tienen en cuenta a la vez los tres mecanismos de agilización de la aplicación de MTT. En contrapartida, el uso de un modelo de observación sencillo como el propuesto en la tesis, complica la tarea de asociación, innecesaria si se emplean modelos de observación específicos. En esos casos, es habitual el uso de métodos probabilísticos de asociación como el JPDA que, debido a su coste computacional (ver en el capítulo 2), imposibilitan también la ejecución del seguidor multi-objeto en aplicaciones de más de tres o cuatro targets; o la complementación de procesos de asociación más simples con otras técnicas como el “*principio de exclusión*” descrito en [MacCormick99].

En esta tesis, la elección de un método de asociación determinístico y sencillo como el NN, elimina los problemas de carga computacional mencionados, y el uso del clasificador de medidas expuesto en la Figura 3.1 robustece la funcionalidad del NN debilitada por su simplicidad, incluso con modelos de observación sencillos como el propuesto.

Finalmente, a continuación se describe también de forma resumida la funcionalidad del clasificador en las distintas etapas del XPFCP, descripción que se completa en el capítulo 6, una vez explicado en el capítulo 5 el funcionamiento de los segmentadores propuestos en la tesis:

1. *Paso de re-inicialización:* Tal y como se explica en el punto anterior, esta etapa añadida a las del PF básico, permite incorporar nuevas hipótesis de seguimiento en la creencia $p(\vec{x}_{t-1}|\vec{y}_{1:t-1})$ generada por el filtro en su ejecución en $t-1$, obteniéndose así un nuevo valor para ésta ($\hat{p}(\vec{x}_{t-1}|\vec{y}_{1:t-1})$), a través de la expresión <3.1>, sobre el que se ejecutan el resto de pasos del XPFCP en el instante t . Como se comenta en párrafos anteriores, la funcionalidad de este paso consiste en añadir un conjunto de partículas al set S_{t-1} que caracteriza a la creencia $p(\vec{x}_{t-1}|\vec{y}_{1:t-1})$.

Para que esta PDF nueva represente de forma robusta a las distintas hipótesis, las muestras a añadir se obtienen de las clases de medidas generadas por el segmentador en el instante $t-1$, $G_{j,t-1|in} / j=1:k_{in,t-1}$. De este modo se evitan los problemas de robustez que aparecen en el XPF ([Koller-Meier01]) y que son analizados en detalle en el capítulo 4.

2. *Paso de corrección*: Esta etapa es modificada fundamentalmente por el proceso de asociación propuesto, basado también en la salida del clasificador de medidas $G_{j,t|in} / j=1:k_{in,t}$. El algoritmo de asociación implementado en el XPFCP, cuyo funcionamiento se explica en detalle en capítulos posteriores, es de tipo NN. Tal y como se comenta en el capítulo 2, el NN es un método determinístico poco fiable, pues permiten obtener únicamente una solución subóptima de asociación, que puede no ser la más correcta sobre todo si se trabaja con modelos simples como el planteado en la tesis. Sin embargo, la baja carga computacional de este tipo de sistemas los hace muy atractivos en aplicaciones sujetas a una especificación de tiempo real como la de interés. Debido a éste último factor, en la tesis se elige usar un algoritmo de asociación NN robustecido mediante el clasificador de medidas.

Para ello, la verosimilitud que se usa en el paso de corrección del PF $p(\bar{g}_{1:k,t|in} | \bar{x}_t)$ se genera a partir de los centroides $\bar{g}_{1:k,t|in}$ obtenidos por este segmentador de medidas, pues estos representan de una forma más robusta al set Y_t . Además, como se explica en el capítulo 5, estos clasificadores incluyen de forma implícita una asociación entre el conjunto de medidas que agrupa cada clase y el hipotético objeto del que éstas se extraen. Esta asociación implícita se completa con la desarrollada en la etapa de corrección, mejorándose así aún más su robustez.

De este modo, la etapa de corrección del XPFCP queda descrita con la ecuación <3.2>, ya expuesta en la Figura 3.1. En capítulos posteriores se analizan en detalle los efectos que tiene el uso de esta nueva PDF de verosimilitud $p(\bar{g}_{1:k,t|in} | \bar{x}_t)$ en la funcionalidad del PF, así como la especificación concreta del algoritmo NN en su aplicación al XPFCP.

$$p'(\bar{x}_t | \bar{y}_{1:t}) = \eta \cdot p(\bar{g}_{1:k,t|in} | \bar{x}_t) \cdot p(\bar{x}_t | \bar{y}_{1:t-1}) \quad <3.2>$$

En este documento se demuestra que la combinación de la funcionalidad determinística del clasificador presentado en este punto con la probabilística aportada por el PF permite alcanzar el objetivo de MTT perseguido en la tesis, cumpliendo además las especificaciones impuestas por la aplicación del algoritmo al área de la robótica personal.

CAPÍTULO 4

CONTENIDO PROBABILÍSTICO DE LA PROPUESTA

En capítulos anteriores se han expuesto los objetivos del trabajo de la tesis y se han analizado proyectos anteriores desarrollados por la comunidad científica en el área de interés. Con todo ello, en el capítulo 3 se concluye la necesidad de utilizar un algoritmo probabilístico para resolver el problema de seguimiento de múltiples objetos desde una plataforma móvil y a través de medidas del entorno realizadas mediante sensores de visión. Se presenta, además, el PF como el estimador probabilístico más adecuado para el desarrollo de la tarea de tracking.

En este capítulo se describen en detalle la funcionalidad y la formalización matemática de este estimador bayesiano. Además, se explican las distintas versiones y modificaciones al algoritmo que, a lo largo de los últimos diez años, distintos autores han aportado con el fin general de robustecer, agilizar e incrementar la fiabilidad del filtro. El fin del análisis expuesto es el de localizar los puntos fuertes y débiles de las distintas versiones del filtro de partículas, para ser respectivamente, aprovechados o reforzados en la propuesta planteada en la tesis.

Al final del capítulo se presenta la versión del PF (el “*PF extendido*”, XPF), a partir de la que se diseña el algoritmo XPFCP propuesto, tal y como se expone en el capítulo anterior. A partir de los estudios analizados a lo largo del capítulo se examinan las características del XPF, con objeto de extraer las necesidades que éste tiene para alcanzar las especificaciones planteadas para el seguidor multimodal de múltiples targets en entornos complejos.

4.1. El filtro de partículas básico. El algoritmo SIR (“*Sequential Importance Resampling*”)

El “*filtro de partículas*” (“*Particle Filter*”, PF) es un estimador recursivo probabilístico que basa su funcionamiento en la representación discreta de la función de creencia a estimar $p(\bar{x}_t | \bar{y}_{1:t})$ por medio de un conjunto de n partículas: $S_t = \{\bar{s}_{i,t} = \bar{x}_t^{(i)} / i=1:n\}$. Cada partícula tiene asociado un peso normalizado que caracteriza su probabilidad de certidumbre dentro de la PDF global: $\bar{w}_t = \{\tilde{w}_t^{(i)} / i=1:n\}$. De este modo se conforma una creencia discreta que es la salida final del algoritmo de estimación:

$$p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t}) \cong S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n \quad <4.1>$$

De esta función de creencia se extrae el valor final de estimación, como se explica en el capítulo anterior, bien mediante la envolvente de la PDF discreta $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$, que caracteriza a la creencia continua $p(\bar{x}_t | \bar{y}_{1:t})$, o bien a través de distintos estadísticos de la creencia discreta como el valor más probable (la mediana) o el valor medio, junto con el valor de varianza, que informa de la fiabilidad del valor de estimación final.

Tal y como se expone ya en el capítulo 2, el PF se corresponde con una de las formulas de implementación en tiempo real del filtro de Bayes. Más concretamente, en esta fórmula se evita la discretización del espacio de estado, o una linealización del modelo del sistema utilizando, como ya se ha descrito, una representación discreta basada en muestras ponderadas de $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t}) \cong S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$. De este modo, se obtienen resultados de estimación más exactos que con otros algoritmos como el KF, que permite obtener formulación teórica correcta del filtro de Bayes, a costa de utilizar un modelo de estado aproximado, o que los algoritmos basados en rejilla, cuyo uso requiere de una discretización del espacio de estado.

La discretización de la creencia a través de un conjunto de muestras ponderadas $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t}) \cong S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ se realiza utilizando la teoría de muestreo de Monte-Carlo, cuya base matemática se comenta en los párrafos siguientes. Debido a ello el filtro toma el nombre de “*Muestreo Secuencial Ponderado*” (“*Sequential Importance Sampling*”, SIS), “*Simulación de Monte-Carlo*” (“*Monte-Carlo Simulation*”) o estimación SMC, “*estimación Secuencial de Monte-Carlo*” (“*Sequential Monte-Carlo estimation*”).

4.1.1. Muestreo de Monte-Carlo

Los primeros ejemplos del algoritmo PF aparecen a mediados de la década de 1950, en literatura especializada de las áreas de física y matemáticas ([Hammersley54]), y en la década siguiente, en el área de control ([Handschin69]). La propuesta inicial consiste en la aplicación de los métodos de Monte-Carlo con el objetivo de obtener una forma muestreada del filtro de Bayes, y más concretamente de la creencia $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$.

El muestreo de Monte-Carlo consiste en aproximar una PDF mediante un conjunto de n muestras, de la siguiente forma:

$$p(\bar{x}_{0:t} | \bar{y}_{1:t}) \equiv \frac{1}{n} \sum_{i=1}^n \delta(\bar{x}_{0:t}^{(i)}), \quad <4.2>$$

donde $S_t = \{\bar{s}_{i,t} = \bar{x}_t^{(i)} / i = 1 : n\}$ son muestras independientes e idénticamente distribuidas (i.i.d.) de la densidad continua, y δ es la función Delta de Dirac.

De acuerdo con la ley de los números grandes, la media estadística de una función de estimación $g(\bar{x}_t)$ del comportamiento de un sistema, obtenida a partir de la PDF a posteriori del mismo $p(\bar{x}_{0:t} | \bar{y}_{1:t})$, o a partir de un conjunto de muestras i.i.d. de $g(\bar{x}_t)$, coincide conforme el número de partículas o muestras n aumenta:

$$E(g(\bar{x}_t)) = \int g(\bar{x}_t) \cdot p(\bar{x}_{0:t} | \bar{y}_{1:t}) \cdot \partial \bar{x}_t \equiv \frac{1}{n} \sum_{i=1}^n g(\bar{x}_{0:t}^{(i)}) \quad <4.3>$$

El problema, es que, tal y como se comenta en el capítulo 2, la probabilidad a posteriori total $p(\bar{x}_{0:t} | \bar{y}_{1:t})$ de la estimación y su marginal, definido allí como creencia $p(\bar{x}_t | \bar{y}_{1:t})$, no se puede calcular para un sistema caracterizado por un espacio de estado infinito. De hecho, el PF surge como solución a este inconveniente. Debido a ello, el valor medio de la probabilidad de estimación se desarrolla a través de una función propuesta como de “*aproximación a la creencia*”, que suele referirse en la literatura con la notación $q(\bar{x}_{0:t} | \bar{y}_{1:t})$. Utilizando esta función y el teorema de Bayes, el valor medio mostrado en la anterior expresión <4.3> queda:

$$\begin{aligned} E(g(\bar{x}_{0:t})) &= \int g(\bar{x}_{0:t}) \cdot p(\bar{x}_{0:t} | \bar{y}_{1:t}) \cdot \partial \bar{x}_t = \int g(\bar{x}_{0:t}) \cdot \frac{p(\bar{x}_{0:t} | \bar{y}_{1:t})}{q(\bar{x}_{0:t} | \bar{y}_{1:t})} \cdot q(\bar{x}_{0:t} | \bar{y}_{1:t}) \cdot \partial \bar{x}_t = \\ &= \int g(\bar{x}_{0:t}) \cdot \frac{p(\bar{y}_{1:t} | \bar{x}_{0:t}) \cdot p(\bar{x}_{0:t})}{p(\bar{y}_{1:t}) \cdot q(\bar{x}_{0:t} | \bar{y}_{1:t})} \cdot q(\bar{x}_{0:t} | \bar{y}_{1:t}) \cdot \partial \bar{x}_t = \int g(\bar{x}_{0:t}) \cdot \frac{w(\bar{x}_{0:t})}{p(\bar{y}_{1:t})} \cdot q(\bar{x}_{0:t} | \bar{y}_{1:t}) \cdot \partial \bar{x}_t \end{aligned} \quad , <4.4>$$

donde $w(\bar{x}_{0:t})$ es la “*función de ponderación de las muestras*” sin normalizar, lo que da lugar al conocido término inglés “*importance sampling function*”:

$$w(\bar{x}_{0:t}) = \frac{p(\bar{y}_{1:t} | \bar{x}_{0:t}) \cdot p(\bar{x}_{0:t})}{q(\bar{x}_{0:t} | \bar{y}_{1:t})} \quad <4.5>$$

Se desarrolla el denominador de la ecuación <4.4> para poder luego normalizar la función de ponderación de las muestras $w(\bar{x}_{0:t})$:

$$\begin{aligned} E(g(\bar{x}_{0:t})) &= \int g(\bar{x}_{0:t}) \cdot \frac{w(\bar{x}_{0:t})}{p(\bar{y}_{1:t})} \cdot q(\bar{x}_{0:t} | \bar{y}_{1:t}) \cdot \partial \bar{x}_t = \int \frac{g(\bar{x}_{0:t}) \cdot w(\bar{x}_{0:t}) \cdot q(\bar{x}_{0:t} | \bar{y}_{1:t})}{p(\bar{y}_{1:t} | \bar{x}_{0:t}) \cdot p(\bar{x}_{0:t}) \cdot \frac{q(\bar{x}_{0:t} | \bar{y}_{1:t})}{q(\bar{x}_{0:t} | \bar{y}_{1:t})}} \cdot \partial \bar{x}_t = \\ &= \int \frac{g(\bar{x}_{0:t}) \cdot w(\bar{x}_{0:t}) \cdot q(\bar{x}_{0:t} | \bar{y}_{1:t})}{w(\bar{x}_{0:t}) \cdot q(\bar{x}_{0:t} | \bar{y}_{1:t})} \cdot \partial \bar{x}_t = \frac{\int g(\bar{x}_{0:t}) \cdot w(\bar{x}_{0:t}) \cdot q(\bar{x}_{0:t} | \bar{y}_{1:t}) \cdot \partial \bar{x}_t}{\int w(\bar{x}_{0:t}) \cdot q(\bar{x}_{0:t} | \bar{y}_{1:t}) \cdot \partial \bar{x}_t} = \end{aligned} \quad <4.6>$$

A la expresión anterior, se le aplica finalmente el muestreo de Monte-Carlo y la ley de números grandes mostrada en la ecuación <4.3>, obteniéndose la siguiente fórmula, que permite calcular el valor medio de la estimación muestreada:

$$E(g(\bar{x}_{0:t})) \equiv \frac{1/n \sum_{i=1}^n g(\bar{x}_{0:t}^{(i)}) \cdot \tilde{w}_t^{(i)}}{1/n \sum_{i=1}^n \tilde{w}_t^{(i)}} = \sum_{i=1}^n g(\bar{x}_{0:t}^{(i)}) \cdot \tilde{w}_t^{(i)}, \quad <4.7>$$

donde $\tilde{w}_t^{(i)}$ representa al vector discreto y normalizado de los pesos que ponderan al conjunto de partículas del PF (a través del muestreo de Monte-Carlo) $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$:

$$E(g(\bar{x}_{0:t})) = \int g(\bar{x}_{0:t}) \cdot \frac{w(\bar{x}_{0:t})}{p(\bar{y}_{1:t})} \cdot q(\bar{x}_{0:t} | \bar{y}_{1:t}) \cdot \delta \bar{x}_t \cong \sum_{i=1}^n g(\bar{x}_{0:t}^{(i)}) \cdot \tilde{w}_t^{(i)}, \quad <4.8>$$

y

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^n w_t^{(i)}} \quad <4.9>$$

A partir de la expresión <4.7> es posible obtener una expresión discreta de la creencia mediante muestreo de Monte-Carlo <4.2> a través de los pesos $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$, así:

$$p(\bar{x}_{0:t}, \bar{y}_{1:t}) \cong \sum_{i=1}^n \tilde{w}_t^{(i)} \cdot \delta(\bar{x}_{0:t}^{(i)}) \quad <4.10>$$

Teniendo en cuenta, además, que se desea conocer únicamente el valor marginal final de la creencia ($p(\bar{x}_t | \bar{y}_{1:t}) \cong \sum_{i=1}^n \tilde{w}_t^{(i)} \cdot \delta(\bar{x}_t^{(i)})$), la ecuación de obtención de los pesos <4.5> se transforma a una forma recursiva, con el objetivo principal de minimizar el uso de recursos de memoria. Para ello se aplica la condición de Markov y los modelos de estado $p(\bar{x}_t | \bar{x}_{t-1})$ y de medidas $p(\bar{y}_t | \bar{x}_t)$ al cálculo de las siguientes funciones de probabilidad, que se desarrollan, además, a través de la ley de probabilidad total:

$$p(\bar{x}_{0:t}) = p(\bar{x}_0) \prod_{j=1}^t p(\bar{x}_j | \bar{x}_{j-1}) = p(\bar{x}_t | \bar{x}_{t-1}) \cdot p(\bar{x}_{0:t-1}), \quad <4.11>$$

$$p(\bar{y}_{1:t} | \bar{x}_{0:t}) = \prod_{j=1}^t p(\bar{y}_j | \bar{x}_j) = p(\bar{y}_t | \bar{x}_t) \cdot p(\bar{y}_{1:t-1} | \bar{x}_{0:t-1}). \quad <4.12>$$

Por otro lado se desarrolla también la forma total de la densidad de aproximación a la creencia, de este modo:

$$q(\bar{x}_{0:t} | \bar{y}_{1:t}) = q(\bar{x}_{0:t-1} | \bar{y}_{1:t-1}) \cdot q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t}) \quad <4.13>$$

Con estas nuevas expresiones, se reescribe la ecuación <4.5>, para lo cual se considera que:

$$w(\bar{x}_{0:t-1}) = \frac{p(\bar{y}_{1:t-1} | \bar{x}_{0:t-1}) \cdot p(\bar{x}_{0:t-1})}{q(\bar{x}_{0:t-1} | \bar{y}_{1:t-1})} \quad <4.14>$$

El resultado es la forma recursiva para el cálculo de los pesos del PF:

$$\begin{aligned} w(\bar{x}_{0:t}) &= \frac{p(\bar{y}_{1:t} | \bar{x}_{0:t}) \cdot p(\bar{x}_{0:t})}{q(\bar{x}_{0:t} | \bar{y}_{1:t})} = \frac{p(\bar{y}_{1:t} | \bar{x}_{0:t}) \cdot p(\bar{x}_{0:t})}{q(\bar{x}_{0:t-1} | \bar{y}_{1:t-1}) \cdot q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})} = \\ &= w(\bar{x}_{0:t-1}) \cdot \frac{p(\bar{y}_{1:t} | \bar{x}_{0:t}) \cdot p(\bar{x}_{0:t})}{p(\bar{y}_{1:t-1} | \bar{x}_{0:t-1}) \cdot p(\bar{x}_{0:t-1}) \cdot q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})} = \\ &= w(\bar{x}_{0:t-1}) \cdot \frac{p(\bar{y}_t | \bar{x}_t) \cdot p(\bar{x}_t | \bar{x}_{t-1})}{q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})} \end{aligned} \quad <4.15>$$

Con la expresión anterior, se calcula de forma iterativa el peso de las muestras i.i.d. $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ que caracterizarán a la creencia, a través de la función de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$.

Tal y como se observa en la expresión <4.15>, la función $w(\bar{x}_{0:t})$ se obtiene, además, a través de la verosimilitud $p(\bar{y}_t | \bar{x}_t)$ (modelo de observación o medida) y del modelo de actuación $p(\bar{x}_t | \bar{x}_{t-1})$ del sistema bajo estudio.

De este modo, y resumiendo todo lo anterior, el muestreo de Monte-Carlo permite obtener una fórmula discreta de la creencia $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$ (semejante a la ya mostrada en la ecuación <4.10>) a partir del conjunto de partículas ponderadas $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$, como:

$$p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t}) = \sum_{i=1}^n \tilde{w}_t^{(i)} \cdot \delta(\bar{x}_t^{(i)}), \quad <4.16>$$

donde el valor de los pesos $\bar{w}_t = [\tilde{w}_t^{(1:n)}]$ se obtiene a partir de $w(\bar{x}_{0:t})$, que se calcula de forma recursiva mediante $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$, y los modelos de observación $p(\bar{y}_t | \bar{x}_t)$ y de estado $p(\bar{x}_t | \bar{x}_{t-1})$ del sistema, de la siguiente forma (ya mostrada en la ecuación anterior <4.15>):

$$w(\bar{x}_{0:t}) = w(\bar{x}_{0:t-1}) \cdot \frac{p(\bar{y}_t | \bar{x}_t) \cdot p(\bar{x}_t | \bar{x}_{t-1})}{q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})}, \quad <4.17>$$

y después se normaliza a través de la siguiente expresión (repetición de la anterior <4.9>):

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^n w_t^{(i)}} \quad <4.18>$$

4.1.2. Primera versión del filtro de partículas: el algoritmo SIS (“*Sequential Importance Sampling*”)

Las expresiones presentadas en el apartado anterior pueden ser utilizadas para desarrollar la forma analítica del filtro bayesiano recursivo a través de las dos etapas típicas de cualquier estimador recursivo (tal y como se explica en el capítulo 2, o en [Gordon93], [Merwe01], [Doucet01], [Arulampalam02] o [Ristic04]). La aplicación del esquema recursivo que caracteriza al filtro de Bayes y permite obtener la forma muestreada de las PDFs implicadas en el proceso de estimación, da lugar al PF:

1. *Predicción*: Cada una de las partículas se propaga al siguiente instante temporal a través del modelo de estado $p(\bar{x}_t | \bar{x}_{t-1})$, para obtener el “valor a priori de la creencia” $p(\bar{x}_t | \bar{y}_{1:t-1})$, también descrita mediante un conjunto de muestras i.i.d. $S_{t|t-1} = \{\bar{s}_{i,t|t-1} = \bar{x}_{t|t-1}^{(i)} / i = 1:n\}$ manteniendo su peso $S_{t|t-1} = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_{t-1}^{(i)}\}_{i=1}^n$, con la expresión:

$$\bar{x}_{t|t-1}^{(i)} = p(\bar{x}_t | \bar{x}_{t-1}) \cdot \bar{x}_{t-1}^{(i)} = f(\bar{x}_{t-1}^{(i)}, \bar{u}_{t-1}, \bar{v}_{t-1}), \quad <4.19>$$

que sustituye a la más genérica <2.23> del capítulo 2, y que se repite aquí por comodidad:

$$p(\bar{x}_t | \bar{y}_{1:t-1}) = \sum_{\forall \bar{x}} p(\bar{x}_t^{(i)} | \bar{x}_{t-1}^{(i)}) \cdot p(\bar{x}_{t-1}^{(i)} | \bar{y}_{1:t-1}) \quad <4.20>$$

2. *Corrección o “ponderación de muestras”* (“*Importance sampling*”): A partir de la versión muestreada de la PDF a priori calculada en la etapa anterior $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t-1})$, se utiliza el muestreo de Monte-Carlo para obtener “la PDF a posteriori” $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$ a través de la expresión <4.16>, la cual sustituye de este modo a la ecuación de corrección del filtro de Bayes (<2.18>) mostrada en el capítulo 2, que se repite aquí por claridad:

$$p(\bar{x}_t | \bar{y}_{1:t}) = \eta \cdot p(\bar{y}_t | \bar{x}_t) \cdot p(\bar{x}_t | \bar{y}_{1:t-1}) \quad <4.21>$$

El cálculo de esta ecuación se sustituye, tal y como se explica en puntos anteriores, por el de los pesos de las muestras $\bar{w}_t = \{w_t^{(i)} / i = 1:n\}$ que caracterizan a la creencia $p(\bar{x}_t | \bar{y}_{1:t})$, a través de las expresiones anteriores <4.17> y <4.18>, mientras que el valor de las partículas no se modifica en este paso $S_{t|t-1} = \{\bar{s}_{i,t|t-1} = \bar{x}_{t|t-1}^{(i)} / i = 1:n\}$. De este modo se completa la definición del set de partículas $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ que describen a la creencia continua $p(\bar{x}_t | \bar{y}_{1:t})$. Debido a su funcionalidad, la etapa de corrección del filtro de Bayes en el PF se denomina también “*importance sampling step*” o, lo que es lo mismo, “*paso de ponderación de muestras*” o “*muestreo ponderado*”.

El objetivo de este paso es el de obtener un peso $\tilde{w}_t^{(i)}$ para cada partícula $\bar{x}_t^{(i)}$ que valore la probabilidad de acierto de ésta como representante del proceso de estimación.

El algoritmo resultante es la forma más básica del PF, que suele aparecer en la literatura científica relacionado al nombre de SIS: “*Sequential Importance Sampling*”, o “*Muestreo Secuencial Ponderado*”.

En los siguientes apartados se muestra como esta versión inicial del PF, si bien es analíticamente correcta, no funciona de forma recursiva más allá de las primeras iteraciones debido a un “*problema de degeneración del set*” ([Doucet97]). Por esta razón el PF no tiene una aplicación muy importante en tareas de estimación hasta que en el año 1993, en el artículo [Gordon93] se propone una solución a este problema.

La degeneración de S_t es consecuencia directa de la ponderación de las partículas, pues con la evolución del tiempo la mayor parte de ellas termina por tener un peso nulo $\tilde{w}_t^{(i)} = 0$, concentrándose el peso total de S_t en unas pocas partículas.

Un razonamiento diferente expuesto en [Kong94] lleva a la misma conclusión. En dicho artículo se demuestra a partir de la ecuación <4.5>, que la varianza de \bar{w}_t ($\Omega^2(w(\bar{x}_{0:t}))$) se incrementa con su evolución en el tiempo. Este hecho llevado al límite temporal, provoca que pasado cierto tiempo, una partícula $\bar{x}_t^{(i1)}$ acabe con un peso asociado de valor unitario $w_{i1} = 1$, mientras que el resto $\{\bar{x}_t^{(i2)} / i2 = 1 : n - i1\}$ finalicen con un valor nulo de su peso $w_{i2} = 0$.

Este problema de degeneración del set de partículas tiene como principal inconveniente la ineficiencia computacional del set $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ degenerado, y en consecuencia del PF, pues las muestras de peso nulo no aportan información a la estimación.

El tema de la degeneración, comentado en párrafos anteriores, ha sido y sigue siendo una de las causas más importantes del desarrollo de distintas versiones del algoritmo SIS básico, que tienen como fin común el diseño de una versión muestreada y fácilmente implementable del filtro de Bayes de eficiencia contrastada. Para valorar las distintas propuestas al algoritmo SIS, surgidas a raíz de la degeneración de S_t , en [Liu98] se define el concepto de “*número de partículas eficaces*” o n_{eff} , de la siguiente forma:

$$n_{eff} = \frac{n}{1 + \Omega^2(w_{opt}(\bar{x}_{0:t}))}, \quad <4.22>$$

donde $\Omega^2(w_{opt}(\bar{x}_{0:t}))$ referencia a la varianza de la función de ponderación óptima, que se obtiene a través de la densidad de aproximación a la creencia óptima:

$$w_{opt}(\bar{x}_{0:t}) = w_{opt}(\bar{x}_{0:t-1}) \cdot \frac{p(\bar{y}_t | \bar{x}_t) \cdot p(\bar{x}_t | \bar{x}_{t-1})}{q_{opt}(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})} \quad <4.23>$$

Desde un punto de vista teórico, el mejor valor para la función de aproximación a la creencia $q_{opt}(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ es aquel que minimice la varianza de \bar{w}_t ([Doucet97]). Según Doucet demuestra en su tesis doctoral, la PDF que cumple este requisito es

$q_{opt}(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t}) = p(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$. A través de esta función sería posible, por tanto, obtener el valor óptimo del vector de pesos $\bar{w}_{opt,t}$. La especificación de minimizar la varianza de \bar{w}_t se usa también a la hora de diseñar otras etapas del PF, como se explica más adelante.

De nuevo, por tanto, el análisis matemático, en este caso del valor de n_{eff} , desemboca en el requerimiento de la creencia $p(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ (idéntica a la expresión de la creencia usada en este documento $p(\bar{x}_t | \bar{y}_{1:t})$, si se aplica la condición de Markov), que es justamente la función incógnita objetivo del PF.

Teniendo en cuenta el hecho de que no se puede utilizar el valor óptimo de esta función, la forma de calcular la eficiencia de un PF, es a través del “valor aproximado de número de partículas eficaces” \hat{n}_{eff} , que se calcula con la siguiente expresión ([Doucet97]):

$$\hat{n}_{eff} = \frac{1}{\sum_{i=1}^n (\tilde{w}_t^{(i)})^2} \quad <4.24>$$

El valor \hat{n}_{eff} es también conocido como “valor aproximado de eficiencia del PF” y es utilizado como factor de calidad comparativo de las distintas versiones del PF que aparecen en la literatura. En esta tesis, este valor se utiliza en capítulos posteriores para corroborar también la eficiencia del algoritmo de seguimiento propuesto por la autora.

Tal y como se puede desprender de las ecuaciones <4.24> y <4.17>, la eficiencia del PF dependerá de $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ que se utilice en la implementación final del PF. En secciones posteriores de este capítulo se presentan algunas de las formas más utilizadas para esta función por la comunidad científica.

Sin embargo, la mejor y más utilizada solución al problema de degeneración que sufre S_t es la incorporación de un tercer paso en el esquema funcional recursivo de este estimador bayesiano, cuya funcionalidad se describe en detalle en el apartado siguiente.

4.1.3. El proceso de “resampling”: el algoritmo SIR (“Sequential Importance Resampling”)

La solución más efectiva al problema de degeneración comentado consiste en la inclusión en el algoritmo SIS básico de una tercera “etapa de selección”, también denominada “re-muestreo” o “resampling”¹. Este paso permite eliminar las partículas con un bajo peso asociado y reproducir a cambio aquellas que tienen un peso elevado, es decir, que constituyen hipótesis más probables de la estimación, de forma semejante al proceso de evolución de especies en los algoritmos genéticos.

¹ El término inglés de “resampling” está ampliamente extendido en la literatura de lengua española relacionada con el PF. Debido a este hecho, en la tesis se utilizará el término inglés para referenciar al proceso de selección incluido en el PF que se explica en los párrafos siguientes.

La inserción de este tercer paso en el algoritmo SIS da lugar al algoritmo SIR: “*Sequential Importance Resampling*”, que traducido como “*Muestreo Secuencial Ponderado con Selección*”.

La formalización matemática de la nueva etapa de resampling sería la siguiente:

1. A partir de $S'_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ ² obtenido a la salida del paso de corrección del PF, repetir n_i veces cada partícula $\bar{x}_t^{(i)}$, de modo que el número de partículas de salida n_i que le corresponde a cada partícula $\bar{x}_t^{(i)}$ esté relacionado con su peso $\tilde{w}_t^{(i)}$, mediante la expresión:

$$E\{n_i\} = n \cdot \tilde{w}_t^{(i)} \Rightarrow \sum_{i=1}^n n_i = n \quad \langle 4.25 \rangle$$

2. A la salida de este paso se obtiene, de este modo, un conjunto de partículas $S_t = \{\bar{s}_{i,t} = \bar{x}_t^{(i)} / i = 1:n\}$, cada una de ellas con un peso idéntico e igual a $\tilde{w}_t^{(1:n)} = n^{-1}$:

$$S_t = \{\bar{x}_t^{(i)}, 1/n\}_{i=1}^n \quad \langle 4.26 \rangle$$

La condición establecida en la ecuación <4.25> para elegir el número de veces n_i a repetir de cada partícula, no es obligatoria tal y como se demuestra en [Kitagawa96]. Debido a ello, gran cantidad de proposiciones de algoritmos de re-muestreo diferentes han sido desarrolladas por distintos grupos de investigación hasta la actualidad ([Merwe01], [Doucet01], [Ristic04], [Douc05] o [Hol06]). En algunos casos el objetivo de la nueva propuesta es el de minimizar la carga computacional de esta nueva etapa en el PF, y en otros la especificación establecida en [Doucet97] de disminuir la varianza de la distribución generada por el conjunto de salida $S_t = \{\bar{x}_t^{(i)}, 1/n\}_{i=1}^n$.

El valor aproximado del número de partículas eficaces \hat{n}_{eff} , presentado en párrafos anteriores, es utilizado en algunas de las técnicas de selección propuestas por distintos autores ([Hue02], [Gustafsson02] o [Kreucher05]) para definir el número de partículas de distinto valor que debe haber en $S_t = \{\bar{x}_t^{(i)}, 1/n\}_{i=1}^n$ tras el paso de resampling, como se explica más adelante.

En todos los casos, el fin principal de cualquier algoritmo de re-muestreo es el de resolver el problema de degeneración comentado, además, tal y como se demuestra en [Merwe01], y en las pruebas comparativas incluidas en el capítulo 6, el uso de una u otra opción no reporta una diferencia importante en eficiencia de la estimación final. Por esta razón, la importancia de la elección de una u otra alternativa para la resolución de la etapa de selección es mínima.

² Se cambia aquí la nomenclatura del set S_t que se obtiene a la salida del paso de corrección, tal y como se explica en el apartado anterior, a S'_t , pues al añadir el paso de selección, el set que se genera a la salida del paso de corrección no representa a la creencia, sino a una PDF aproximada a ésta que se denomina a partir de aquí $p'(\bar{x}_t | \bar{y}_{1:t})$. La salida del nuevo paso de selección es entonces la que caracteriza a la creencia $p(\bar{x}_t | \bar{y}_{1:t})$, que adopta así el nombre S_t .

En este capítulo se describen los algoritmos de resampling más utilizados por la comunidad científica, para analizar su idoneidad a los objetivos propuestos en la tesis.

Si bien el PF, debido a los problemas de degeneración, no fue usado en aplicaciones de estimación hasta la introducción del ya mencionado paso de selección en [Gordon93], la inserción de esta nueva etapa en el filtro de Bayes produjo nuevas complicaciones: debido justamente al proceso de selección, todas las partículas de S_t que caracteriza a la creencia $p(\bar{x}_t | \bar{y}_{1:t})$ al final de los tres pasos del PF pueden terminar concentrándose en una misma hipótesis, esto es: $n_{i1} = n$ para una única partícula $\bar{x}_t^{(i1)}$ mientras que $n_{i2} = 0$ para el resto de partículas $\{\bar{x}_t^{(i2)} / i2 = 1 : n - i1\}$. Este hecho hace que el PF pierda la diversidad que lo hace versátil, flexible, robusto y por supuesto, multimodal. Este problema, conocido con el nombre de como “*empobrecimiento del set de partículas*” ([Pitt99], [Merwe01], [Doucet01] o [Arulampalam02]), ya se describe en el capítulo anterior, y depende de la función $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ elegida para la etapa de corrección, siendo especialmente relevante si el ruido asociado al modelo de actuación \bar{v}_t , de la ecuación <2.9> (del capítulo 2), es pequeño.

En este capítulo se revisan también distintas técnicas propuestas por la comunidad científica para resolver este nuevo problema de empobrecimiento, que en general, consisten en la introducción de nuevos pasos en el ciclo básico del PF o la modificación de $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$. La idoneidad y eficacia de estos métodos, aplicados a la tarea de MTT, son también analizadas.

En cualquier caso, la solución inicial que se propone en distintos artículos (como en [Gordon97], [Doucet01] o [Gustafsson02]) para resolver el problema de empobrecimiento del set, consiste simplemente en dispersar las partículas de S_t por el espacio de estado a través de un modelo de ruido de estado, que realice la función de diversificación que desarrollaría, en caso de existir, el ruido real \bar{v}_t asociado al modelo de estado del sistema bajo estudio. Esta solución tan sencilla aporta a S_t la diversidad que requiere para evitar su empobrecimiento, y tiene una funcionalidad similar a las que, con una complejidad y carga computacional mayores, son descritas más adelante en este capítulo.

Es importante destacar en este punto que muchas de las técnicas propuestas por distintos autores para solucionar este problema de empobrecimiento conllevan un incremento del coste computacional del PF que las convierte en impracticables para los objetivos aquí expuestos.

Antes de abordar la revisión de cada una de las modificaciones al algoritmo SIR comentadas en párrafos anteriores, en el apartado siguiente se presenta, a modo de resumen, un diagrama funcional con los tres pasos típicos de la versión estándar del PF: el algoritmo SIR.

4.1.4. Diagrama funcional del algoritmo SIR

A partir de lo expuesto hasta el momento se puede concluir el hecho de que la versión básica del PF consta de tres etapas: “*Predicción, Corrección y Selección*”. El funcionamiento recursivo de éstas puede resumirse en los siguientes párrafos:

0. El ciclo de un algoritmo SIR comienza en el instante t con un conjunto de muestras ponderadas por igual e i.i.d. $S_{t-1} = \{\bar{x}_{t-1}^{(i)}, \tilde{w}_{t-1}^{(i)}\}_{i=1}^n = \{\bar{x}_{t-1}^{(i)}, 1/n\}_{i=1}^n$ que representan el valor de la creencia estimada por el PF en el instante $t-1$: $p(\bar{x}_{t-1} | \bar{y}_{1:t-1})$.
1. *En el paso de predicción*, estas partículas son propagadas a través del modelo de estado del sistema bajo estudio $p(\bar{x}_t | \bar{x}_{t-1})$, obteniendo gracias la expresión <4.19>, un nuevo set $S_{t|t-1} = \{\bar{s}_{t|t-1}^{(i)} = \bar{x}_{t|t-1}^{(i)} / i = 1:n\}$ con pesos idénticos $\bar{w}_{t-1} = \{w_{t-1}^{(i)} = 1/n / i = 1:n\}$, que caracteriza la estimación a priori de la creencia $p(\bar{x}_t | \bar{y}_{1:t-1})$. El peso de las partículas no se modifica en esta etapa, quedando por tanto representado el set por $S_{t|t-1} = \{\bar{x}_{t|t-1}^{(i)}, 1/n\}_{i=1}^n$.
2. El peso de las partículas es recalculado *en el paso de corrección* $\bar{w}_t = \{w_t^{(i)} / i = 1:n\}$, utilizando para ello la función de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ y las PDFs que caracterizan al modelo de actuación $p(\bar{x}_t | \bar{x}_{t-1})$ y de observación $p(\bar{y}_t | \bar{x}_t)$, mediante la ecuación recursiva <4.17> que permite calcular la función de ponderación de las muestras $w(\bar{x}_{0:t})$. El set de partículas que se obtiene a la salida de esta etapa será identificado como: $S_t' = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$, y la PDF a la que representa (aproximación del valor final de la creencia $p(\bar{x}_t | \bar{y}_{1:t})$) como: $p'(\bar{x}_t | \bar{y}_{1:t})$.
3. Finalmente, *en el paso de selección*, a través del esquema de resampling elegido y utilizando el vector de pesos calculado en la etapa anterior $\bar{w}_t = \{w_t^{(i)} / i = 1:n\}$, se obtiene el nuevo conjunto de partículas $S_t = \{\bar{x}_t^{(i)}, 1/n\}_{i=1}^n$. Esta es, además, la salida discreta que proporciona el PF como estimación final de la densidad de creencia o PDF a posteriori $p(\bar{x}_t | \bar{y}_{1:t})$ para el instante t , tal y como expresa la ecuación <4.16> anterior.

La Figura 4.1 muestra la funcionalidad del PF basada en tres etapas, mediante un diagrama de bloques. Esta figura tiene una especial relevancia, pues será ampliada más adelante en este capítulo y los sucesivos, conforme se vayan introduciendo distintas modificaciones en el comportamiento básico del filtro probabilístico hasta llegar a la solución multimodal y dinámica propuesta en la tesis. Por otro lado, en la Figura 4.2 se muestra un ejemplo unidimensional ($\bar{x} = x$) de la evolución que sufren las partículas a lo largo de los distintos pasos de ejecución del algoritmo SIR.

Durante la última década, a la vez que el PF se convertía en una herramienta para desarrollar tareas de estimación muy utilizada por investigadores de distintas áreas científicas, muchos partidarios del uso del KF como filtro probabilístico han centrado sus planteamientos en el siguiente hecho, ya comentado en apartados anteriores:

“Para que la representación de la creencia $p(\bar{x}_t | \bar{y}_{1:t})$ en base a muestras ponderadas $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ en la que se soporta el fundamento matemático del PF sea aproximada a la

real, el set tiene que incluir un número n elevado de partículas, hecho que conlleva el incremento de carga computacional que puede invalidar al algoritmo para su aplicación en tareas con restricciones de tiempo real”.

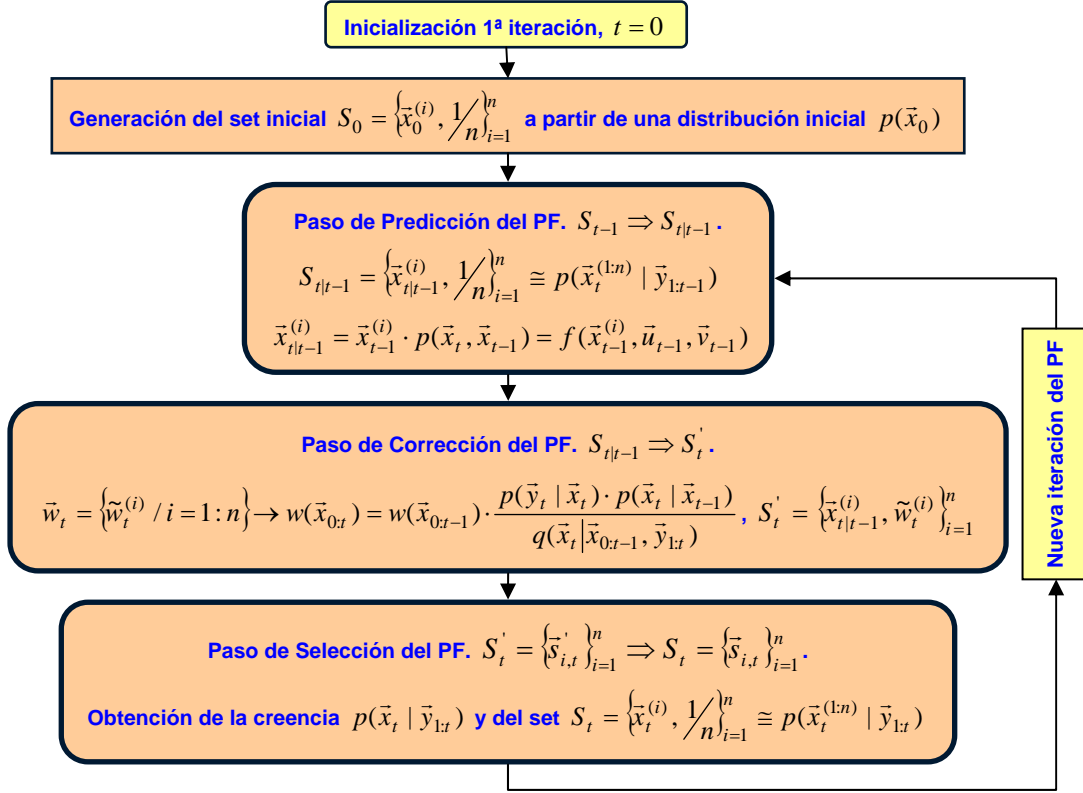


Figura 4.1. Diagrama funcional del algoritmo PF.

En esta tesis se muestra de forma práctica que esta premisa no es necesariamente cierta, pues si se combina un algoritmo determinístico con el PF se aumenta la robustez del mismo y se minimiza el número de partículas necesarias para obtener del mismo una estimación fiable.

En la Figura 4.1, a las etapas que conforman la ejecución del PF ya comentadas, se añade un paso de inicialización que solo se ejecuta en la primera iteración del filtro. Este paso consiste en obtener una versión muestreada de la probabilidad a priori del vector de estado a estimar $p(\bar{x}_0^{(1:n)})$. Existen distintas opciones para resolver este aspecto:

- Teniendo en cuenta que este valor inicial es desconocido, la mayor parte de las aplicaciones del PF a problemas reales de la literatura consultada plantean utilizar como representación de esta distribución inicial $S_0 = \{\bar{x}_0^{(i)}, 1/n\}_{i=1}^n$ una aleatoria y uniforme en el espacio de estado.
- Existen otras propuestas consistentes en elegir como probabilidad a priori inicial una distribución relacionada con la verosimilitud del sistema de interés en ese instante:

$$p(\bar{x}_0) \propto p(\bar{y}_0 | \bar{x}_0) \tag{4.27}$$

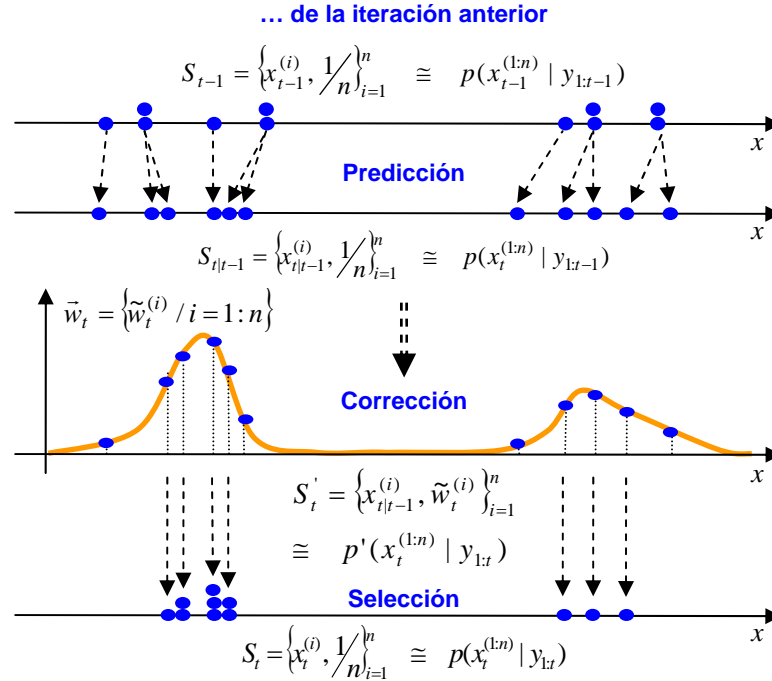


Figura 4.2. Representación unidimensional de la evolución de $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$, donde $n = 11$, en las diferentes etapas del PF. El valor de las partículas en el espacio de estado se representa con puntos azules. La función de “importance sampling” $w(x_{0:t})$ usada en el paso de corrección se muestra en naranja, y los puntos azules sobre ella el valor del peso asignado a cada partícula.

Esta última solución es, ligeramente modificada, la que se plantea usar en esta tesis. El objetivo de la elección es que el filtro bayesiano converja más rápidamente hacia el valor de creencia real, para lo cual, como regla general, es necesario incorporar información de la verosimilitud a la PDF de creencia. En apartados posteriores de este capítulo se pone de manifiesto estas afirmaciones, sobre todo, a la hora de buscar el valor más adecuado para la densidad de aproximación a la creencia.

4.1.5. La versión más extendida del algoritmo SIR: el algoritmo “Bootstrap”

A la hora de diseñar una nueva versión del PF, el valor de la densidad de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ es una de las variables que más puede modificar la calidad del PF resultante, mientras que la técnica de resampling utilizada no supone una diferencia importante en la efectividad del estimador final. Debido a ello, la elección de uno u otro valor para esta función da lugar a muchas de las distintas versiones del PF existentes en la literatura científica.

Desde un punto de vista teórico, como ya se ha comentado en párrafos anteriores, el valor óptimo de esta función es aquel que minimiza la varianza de los pesos: $q_{opt}(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t}) = p(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$. Sin embargo, también se ha comentado anteriormente que esta función no es implementable en la práctica.

A la hora de implementar el PF, la opción más generalizada ([Gordon93], [Isard96], [Kitagawa96], [Liu98], [Doucet00] o [Merwe01]) debido a su sencillez, es la de usar como función de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ la PDF que caracteriza al modelo de actuación del sistema: $p(\bar{x}_t | \bar{x}_{t-1})$. En ese caso, la ecuación <4.17> que permite obtener el valor de los pesos asociados a las partículas en cada ciclo del algoritmo SIR, se simplifica así:

$$\begin{aligned} w(\bar{x}_{0:t}) &= w(\bar{x}_{0:t-1}) \cdot \frac{p(\bar{y}_t | \bar{x}_t) \cdot p(\bar{x}_t | \bar{x}_{t-1})}{q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})} = w(\bar{x}_{0:t-1}) \cdot \frac{p(\bar{y}_t | \bar{x}_t) \cdot p(\bar{x}_t | \bar{x}_{t-1})}{p(\bar{x}_t | \bar{x}_{t-1})} <4.28> \\ &= w(\bar{x}_{0:t-1}) \cdot p(\bar{y}_t | \bar{x}_t) \end{aligned}$$

Esta propuesta para la función $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ aparece por primera vez ya en [Gordon93], publicación en la que, tal y como ya se ha comentado, el autor introduce el paso de selección en el algoritmo SIS, dando lugar al PF básico o algoritmo SIR. Gordon bautiza al estimador resultante con el nombre de “*Bootstrap Filter*”, cuya traducción podría ser “*Filtro Autónomo*”. El nombre elegido por el autor se debe al hecho ya explicado de que, gracias a su contribución al algoritmo SIS, las partículas del algoritmo “*Bootstrap*” se ubican automáticamente en zonas de alta probabilidad del espacio de estado.

El mismo valor de $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ es el propuesto para ser utilizado en el algoritmo “*Condensation*” ([Isard98a]), presentado en el capítulo 2 como uno de los primeros utilizado en aplicaciones de MTT a partir de información de visión. De hecho, el algoritmo “*Bootstrap*” es la versión más usada del PF para resolver este tipo de problemas debido, como ya se ha comentado, a su bajo coste computacional.

La elección de $p(\bar{x}_t | \bar{x}_{t-1})$ como función de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ tiene el inconveniente de eliminar de esta PDF cualquier tipo de información acerca de la verosimilitud $p(\bar{y}_t | \bar{x}_t)$ del modelo implicado en el proceso de estimación. Debido a ello (la Figura 4.3 muestra una representación unidimensional y unimodal del problema, extraída de [Merwe01]), si $p(\bar{y}_t | \bar{x}_t)$ es estrecha o el modelo de actuación $p(\bar{x}_t | \bar{x}_{t-1})$ no es bueno, es posible que la PDF $p(\bar{y}_t | \bar{x}_t)$ se ubique en la cola de la densidad a priori $p(\bar{x}_t | \bar{y}_{1:t-1})$, caracterizada por $S_{t|t-1}$.

Esta situación conduce a la generación de pesos de valor muy bajo para la mayor parte de partículas que representan a la PDF a la salida del paso de corrección, $p'(\bar{x}_t | \bar{y}_{1:t-1})$: $S'_t = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$, pues, tal y como se observa en la ecuación <4.28> anterior, el valor de los pesos $\tilde{w}_t = [\tilde{w}_t^{(1:n)}] \rightarrow w(\bar{x}_{0:t}) = w(\bar{x}_{0:t-1}) \cdot p(\bar{y}_t | \bar{x}_t)$ se obtiene directamente de la relación de cada partícula de $S_{t|t-1}$ que caracteriza a la creencia a priori, con la PDF de verosimilitud, y en la situación descrita en la Figura 4.3, esta relación es muy débil. En este contexto el PF diverge muy rápidamente, debido al problema de empobrecimiento de S'_t .

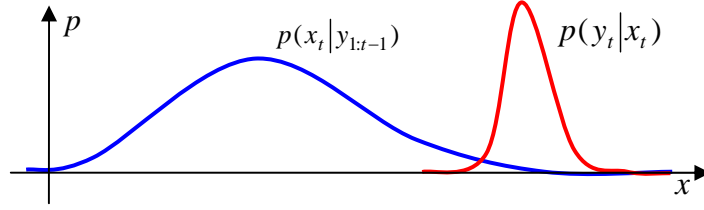


Figura 4.3. Representación unidimensional y unimodal del problema asociado a la elección de una función de aproximación a la creencia igual a $p(\bar{x}_t | \bar{x}_{t-1})$, característica del algoritmo “Bootstrap”. En la figura la creencia a priori $p(\bar{x}_t | \bar{y}_{1:t-1})$ representada por $S_{t|t-1}$ se muestra en azul, y la verosimilitud $p(\bar{y}_t | \bar{x}_t)$ se muestra en rojo.

Por otro lado, la existencia de una diferencia importante entre las funciones $p(\bar{y}_t | \bar{x}_t)$ y $p(\bar{x}_t | \bar{y}_{1:t-1})$, como la mostrada en la Figura 4.3, puede producir la inestabilización del estimador al hacerse muy sensible a ruidos en las medidas \bar{o}_t o al no poder adaptarse todo lo rápidamente que sería deseable a los movimientos del objeto a seguir. Estos problemas, al margen del de empobrecimiento, son importantes si el filtro se usa en tareas de seguimiento como la de interés.

Para resolverlos todos, la comunidad científica propone varias modificaciones para la densidad de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t}) = p(\bar{x}_t | \bar{x}_{t-1})$ implicada en el algoritmo “Bootstrap” que son descritas en el apartado siguiente.

En cualquier caso, en distintas referencias ([Merwe01], [Doucet01] o [Ristic04]) se concluye que, en general, cada autor suele diseñar la propuesta de $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ que considera más adecuada en el contexto en el que pretende utilizar el PF.

La función $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ que se propone en esta tesis se presenta en el capítulo 6, una vez analizadas las características de las alternativas más interesantes desarrolladas por la comunidad científica para esta función. La base de la propuesta de $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ presentada es la misma del algoritmo “Bootstrap” (es decir, $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t}) = p(\bar{x}_t | \bar{x}_{t-1})$), debido sobre todo a su sencillez y baja carga computacional.

Sin embargo, en la propuesta algorítmica planteada en esta tesis, la función de peso $w(x_{0:t})$ resultante al emplear la PDF de actuación $p(\bar{x}_t | \bar{x}_{t-1})$ como función de aproximación, se modifica mediante la salida del proceso de clasificación de medidas $Y_t \Rightarrow G_{1:k,t|in}$ (ver capítulo 3). De este modo, el PF resultante no sufre los problemas de empobrecimiento e inestabilidad característicos del algoritmo “Bootstrap”, como se demuestra en este documento.

4.2. Modificaciones al algoritmo SIR

Durante los años finales de la década de 1990, distintos investigadores fundamentalmente del área de procesamiento de señal, proponen múltiples modificaciones al PF básico ([Doucet01], [Merwe01], [Arulampalam02] o [Ristic04]).

La mayor parte de ellas constituyen una propuesta en alguna de las dos siguientes líneas, y siempre con el fin de mejorar la eficiencia del estimador y/o eliminar el problema de empobrecimiento de S_t que sufre el algoritmo “*Bootstrap*”:

1. *El proceso de selección, o*
2. *la función de aproximación a la creencia: $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$.*

Además, el empobrecimiento del set que aparece con la necesaria introducción del paso de selección en el algoritmo SIS, y que da lugar al algoritmo SIR, puede solucionarse eliminando esa etapa del flujograma de funcionamiento del PF básico, y sustituyéndolo por otro que, con la misma funcionalidad, no produzca el empobrecimiento de S_t que caracteriza a la creencia de salida del PF $p(\bar{x}_t | \bar{y}_{1:t})$.

El análisis de las soluciones planteadas por la comunidad científica en esta última línea se sale del objetivo de este capítulo. Sin embargo, debido al uso extensivo de éstos en varios trabajos de área de interés, los más importantes se describen en el apéndice I.

Una vez analizadas en este punto las alternativas y modificaciones al PF básico más importantes diseñadas por la comunidad científica para mejorar la funcionalidad del filtro, en el apartado siguiente se presenta el algoritmo que se usa como base para diseñar la propuesta de estimación multimodal desarrollada en esta tesis.

4.2.1. El algoritmo de selección

En este apartado se van a presentar algunos de los algoritmos de selección más utilizados en el PF que se pueden encontrar en la literatura científica.

a) “*Re-muestreo Multinomial*”:

El “*Re-muestreo Multinomial*” (“*Multinomial Resampling*”), introducido en [Gordon93], es el primer algoritmo de resampling diseñado para el PF.

El funcionamiento de este algoritmo consiste en muestrear uniformemente n veces la creencia discreta a la salida del paso de corrección del PF $p'(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$, caracterizada por $S_t' = \{\bar{s}_{i,t|t-1}' = \bar{x}_{i,t-1}^{(i)} / i = 1:n\}$, cada una con probabilidad $\bar{w}_t = \{\tilde{w}_t^{(i)} / i = 1:n\}$.

De este modo, las partículas con peso $\tilde{w}_t^{(i)}$ mayor tendrán un número de repeticiones n_i (también llamados “*hijos*”) también mayor, en la creencia discreta de salida del algoritmo de resampling $S_t = \{\bar{x}_t^{(i)}, 1/n\}_{i=1}^n$, y viceversa. Para mantener la uniformidad del set de partículas resultante el re-muestreo ha de ser aleatorio.

Existen distintas implementaciones prácticas de este método de resampling, además de la mostrada en [Gordon93], como la descrita en [Pitt99], si bien su base es siempre la expuesta.

b) “*Re-muestreo Residual*”:

Una técnica alternativa de resampling es presentando en [Liu98]. Este otro esquema es llamado “*Residual Resampling*” o “*Re-muestreo Residual*” por los autores de [Liu98], y es computacionalmente menos costoso que el multinomial.

Debido a su menor coste, este método de selección ha sido el utilizado como base para la implementación del paso de selección en el algoritmo de seguimiento desarrollado en esta tesis, ya que, tal y como ya se ha comentado, el uso de ésta u otra técnica de selección no afecta a la eficacia del sistema de tracking propuesto.

La especificación concreta del proceso de muestreo residual usado en la tesis se define en el capítulo 6, mientras que en aquí se analiza su funcionalidad genérica, dividida en dos fases:

1. En la primera se obtiene de forma determinística el número n_i de “*hijos*” a extraer de cada partícula $\bar{x}_t^{(i)}$ a través del valor normalizado de su peso asociado $\tilde{w}_t^{(i)}$, del siguiente modo:

$$n_i = n \cdot \tilde{w}_t^{(i)} \quad \langle 4.29 \rangle$$

2. En la segunda se reparten el número de muestras que falte por asignar $(n - \sum_{i=1}^n n_i)$ de forma aleatoria entre todo el set de partículas de entrada $S'_t = \{\bar{x}_{t|t-1}^{(i)} / i=1:n\}$, mediante el procedimiento de selección multinomial.

c) Otros:

Existen otros muchos algoritmos de resampling propuestos por distintos autores en la literatura científica del área, cuya finalidad, en la mayor parte de los casos, es, por un lado, disminuir la varianza del conjunto de partículas de salida $S_t = \{x_t^{(i)}, 1/n\}_{i=1}^n$ y, por otro, minimizar la carga computacional del algoritmo resultante.

Algunos de los más extendidos, a parte de los ya expuestos se enumeran a continuación a modo de revisión ([Merwe01], [Doucet01], [Arulampalam02], [Ristic04], [Douc05] o [Hol06]):

- el “*Re-muestreo Sistemático*” (“*Systematic Resampling*”) propuesto en [Carpenter99] y con otras versiones, como la “*residual*” (“*Residual Systematic Resampling*”) en [Bolic03];
- el método “*Tree Based Branching*” propuesto por Crisan en [Doucet01];
- el “*Re-muestreo Estratificado*” (“*Stratified Resampling*”) presentado en [Kitagawa96]; o
- el método de “*Mínima Varianza*” (“*Minimum Variance*”), descrito en [Merwe01] como combinación de los dos anteriores, y que se caracteriza por minimizar la varianza del conjunto de partículas de salida $S_t = \{x_t^{(i)}, 1/n\}_{i=1}^n$ respecto al de entrada $S'_t = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$.

Con objeto de encontrar la alternativa de re-muestreo más adecuada para la aplicación de interés, varias de las aquí expuestas han sido probadas en la propuesta de estimación basada en un PF de esta tesis. Los resultados de dichas pruebas se exponen en el capítulo 6, y de ellos se extrae la misma conclusión ya comentada en párrafos anteriores de que la eficiencia del PF no se ve modificada de forma importante por la técnica de resampling empleada.

4.2.2. La función de aproximación a la creencia

En esta sección se analizan dos tipos de alternativas para resolver el problema de empobrecimiento del set que padece el algoritmo “*Bootstrap*”, en las condiciones comentadas:

1. En los apartados a), b) y c) se describen propuestas que se basan en realizar modificaciones sobre el algoritmo “*Bootstrap*”, manteniendo como función de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ el valor $p(\bar{x}_t | \bar{x}_{t-1})$ que lo representa, y
2. en los apartados d), e) y f) se exponen las propuestas centradas en la *elección de una nueva densidad de aproximación a la creencia* $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$.

El paso de “*re-inicialización*”, presentado en el capítulo anterior y analizado en secciones posteriores, tiene un efecto semejante sobre el empobrecimiento del set al de los procesos que se analizan en los apartados a), b) y c) siguientes. Sin embargo, al contrario que estos últimos, la etapa de re-inicialización no empeora la exactitud de la estimación, y permite además adaptar la creencia $p(\bar{x}_t | \bar{y}_{1:t})$ para que caracterice un número variable de hipótesis de estimación.

En [Rui01] se establece, a su vez, una segunda clasificación a la hora de explicar las distintas alternativas desarrolladas por la comunidad científica para obtener un nuevo valor para la mencionada densidad $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$:

1. *Las técnicas directas*, basadas en la modificación de la propuesta $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t}) = p(\bar{x}_t | \bar{x}_{t-1})$ para incluir en la PDF cierta información relacionada con el vector de medidas Y_t (o la verosimilitud $p(\bar{y}_t | \bar{x}_t)$), con objeto de obtener una forma para esta densidad más similar a la óptima $q_{opt}(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t}) = p(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$. Un ejemplo de estos métodos se describe en el apartado e) de este punto.
2. *Las indirectas*, que se basan en el uso de un filtro complementario que permita hacer un seguimiento del valor de la PDF de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ a lo largo del tiempo. Los estimadores usados más habitualmente en este tipo de propuestas son el KF (ver apartado g) de este punto) y el propio PF (apartado f) de este punto). Los métodos de estimación basados en estas soluciones se denominan también maestro-esclavo, debido a la relación jerárquica que aparece entre los dos filtros implicados en ellos. El algoritmo maestro es, en este caso siempre, el PF, y el esclavo el algoritmo añadido, que permite obtener el valor de $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$.

Si bien se ha demostrado ([Rui01]) que las técnicas indirectas arrojan, en general, mejores resultados que las directas, estas alternativas no son en todos los casos óptimas. Al margen de

que la ejecución del filtro añadido incrementa de forma importante la carga computacional del algoritmo global, no siempre es posible ajustar éste para obtener el valor óptimo de $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ en cada aplicación, tal y como se analiza en los apartados siguientes.

Además, es importante destacar el hecho de que la robustez final de PF en el que se utiliza una técnica indirecta para calcular el valor de $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ depende de la del filtro añadido a través del que se obtenga la mencionada densidad.

a) “Edición Previa” y “Rechazo Selectivo” de partículas:

El procedimiento de “Edición Previa” de partículas (“Prior Editing”) fue introducido en [Gordon93]. El proceso de edición consiste en eliminar las partículas del set $S_{t|t-1} = \{\bar{s}_{i,t|t-1} = \bar{x}_{t|t-1}^{(i)} / i = 1:n\}$ que, tras la etapa de predicción, arrojen una estimación de observación (dada por la expresión <4.30> siguiente, extraída de la <2.11> del capítulo 2) muy diferente de las medidas Y_t .

$$\bar{y}_{t|t-1}^{(i)} = \bar{x}_{t|t-1}^{(i)} \cdot p(\bar{y}_t | \bar{x}_t) = h(\bar{x}_{t|t-1}^{(i)}, \bar{u}_t, \bar{o}_t) \quad <4.30>$$

De esta forma se obtiene a la salida del PF modificado, un nuevo conjunto de partículas \hat{S}_t que representa de forma aproximada al valor de la creencia $p(\hat{\bar{x}}_t | \bar{y}_{1:t})$, y que no sufre el problema de empobrecimiento característico del algoritmo SIR.

Para que el número de partículas n se mantenga constante a lo largo de las distintas iteraciones del PF modificado, las partículas eliminadas en el paso de “Edición Previa” son repuestas en el de resampling. La elección de las partículas a rechazar se hace a través de una serie de parámetros que se fijan en función de cada aplicación concreta, por lo que además de computacionalmente caro, este proceso es poco riguroso.

Un procedimiento más riguroso, pero con el mismo funcionamiento que el de “Edición Previa” es el denominado “Rechazo Selectivo” de partículas (“Selective Rejection”). El proceso de rechazo tiene el mismo fin que el método anterior de modificar $S'_t = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ eliminando algunas de ellas, pero en este caso se propone realizar la eliminación tras el paso de corrección, a la vez que el de selección, que además permite asegurar también en este caso, que n a la salida del filtro es constante.

El procedimiento de “Rechazo Selectivo” es solo aplicable en aquellas tareas de estimación en los que la verosimilitud esté limitada en el espacio del vector de medidas ($p(\bar{y}_t | \bar{x}_t) < \text{lim}$). Su funcionamiento consiste, al igual que el de la “Edición Previa”, en rechazar aquellas partículas que arrojan una estimación de observación (dada por la expresión <4.31>) muy diferente (mayor de un límite lim) al conjunto de medidas real Y_t :

$$\bar{y}_t^{(i)} = \bar{x}_t^{(i)} \cdot p(\bar{y}_t | \bar{x}_t) = h(\bar{x}_t^{(i)}, \bar{u}_t, \bar{o}_t) \quad <4.31>$$

El algoritmo de “*Rechazo Selectivo*” requiere un número aleatorio de iteraciones para su ejecución, tal y como se demuestra en [Pitt99], por lo que, aún teniendo una definición más rigurosa que el de “*Edición Previa*”, es del mismo modo computacionalmente inadecuado para aplicaciones con restricciones de tiempo real, como la de interés.

Además, el inconveniente principal de ambas modificaciones es que el conjunto de partículas \hat{S}_t que se obtiene a la salida del PF transformado pierde exactitud a la hora de representar al valor continuo de la creencia $p(\hat{x}_t | \hat{y}_{1:t})$. Esta inexactitud, además no es siempre fácil de analizar teóricamente, por lo que puede no ser conocida.

Por todo lo anterior, las modificaciones de “*Rechazo Selectivo*” y “*Edición Previa*” no son adecuadas para el estimador multimodal objetivo de la tesis.

b) Paso MCMC (“*Markov Chain Monte-Carlo Step*”):

El paso MCMC (“*Markov Chain Monte-Carlo Step*” o “*MCMC Step*”) es, al igual que el anterior, un algoritmo usado para modificar el valor de las partículas S_t (“*mover*” las partículas a un nuevo valor \hat{S}_t) tras la etapa de selección del PF, obteniéndose de este modo un valor modificado de la creencia a la salida del filtro ([Carpenter99], [Merwe01], [Doucet01], [Arulampalam02] o [Ristic04]).

En este caso, para conseguir el efecto de movimiento se aplica un núcleo de transición o “*movimiento*” basado en una cadena de Markov $K(\hat{x}_{0:t}, \hat{x}_{0:t})$ a S_t una o más veces, tal que:

$$\int K(\hat{x}_{0:t}, \hat{x}_{0:t}) \cdot p(\hat{x}_t | \hat{y}_{1:t}) = p(\hat{x}_t | \hat{y}_{1:t}) \quad <4.32>$$

Alguno de los núcleos de transición $K(\hat{x}_{0:t}, \hat{x}_{0:t})$ más utilizados son el muestreador de Gibbs ([Hue02]) y el de Hastings-Metropolis ([Gilks96] o [Merwe01]).

La ventaja del uso de este método de transformación de S_t , frente a los explicados en el punto anterior, radica en que, si se elige de forma adecuada el núcleo de transición $K(\hat{x}_{0:t}, \hat{x}_{0:t})$, esta transformación no afecta a la exactitud de la estimación del filtro ([Doucet01]).

Su inconveniente principal, al margen del aumento de carga computacional que supone para la ejecución del PF final, es que esta etapa modifica el valor de las partículas del conjunto a \hat{S}_t , de forma no controlada. Además, el paso de MCMC es también iterativo, tal y como se desprende de la ecuación <4.32>, por lo que el proceso de estimación llevado a cabo por el PF puede no converger al añadir esta etapa ([Vermaak05a]).

La Figura 4.4 muestra, a modo de ejemplo, el funcionamiento en una iteración de un PF (\hat{x}_t unidimensional) en el que se ha insertado un paso MCMC. En la figura se observa fácilmente que el objetivo de esta etapa añadida al esquema básico del PF es el de diversificar el valor de

las partículas S_t tras el paso de resampling, con el fin de evitar el empobrecimiento del set, sin alterar, por otro lado, la exactitud de la creencia estimada a la salida del filtro $p(\bar{x}_t^{(i)} | \bar{y}_{1:t})$.

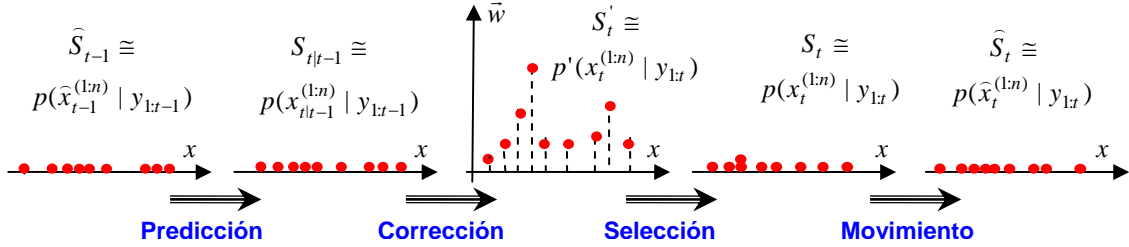


Figura 4.4. Representación funcional de un proceso de “movimiento” genérico de $S_t = \{x_t^{(i)}, 1/n\}_{i=1}^n$, para un \bar{x}_t unidimensional. El valor en el espacio de estado de las partículas y de su peso se muestra en rojo.

El paso MCMC aparece descrito por primera vez en [Liu98], si bien la misma idea, con la misma funcionalidad, exportada a problemas de estimación más complejos es presentada más tarde en [Doucet01] por Berzuini. En este último caso los autores denominan a este paso algoritmo de “Selección y Movimiento” (“Resampling and Moving”).

El nombre de este algoritmo se debe al hecho evidente de que la etapa de re-muestreo se sustituye, en este caso, por otra de “resampling + movimiento” del conjunto de partículas.

Los autores del algoritmo “Resampling and Moving” recomiendan su incorporación al PF cuando el filtro bayesiano se aplica a problemas de identificación del modelo de un sistema inicialmente desconocido. En estos casos, el vector de estado crece ($\bar{x}_t = g(f_1(\bar{x}_{t-1}), f_2(\bar{x}_t^+))$) según se incorporan parámetros nuevos \bar{x}_t^+ al modelo, que tendrán, en general, una dinámica $f_2(\bar{x}_t^+)$ y un espacio de representación, en general, diferente al de los ya incluidos en \bar{x}_t .

En la aplicación de identificación, el carácter estático de los parámetros a estimar implica un valor prácticamente nulo del vector de ruido asociado al de estado \bar{v}_t , y, por lo tanto, hace imprescindible el uso de una etapa de “movimiento” que evite el empobrecimiento del set de partículas del filtro. Además debido a la dinámica múltiple inherente a este tipo de aplicaciones, el paso de “Selección y Movimiento” sirve también para cambiar de espacio de configuración ($f_i(\bar{x}) \Rightarrow f_j(\bar{x})$) la evolución de los parámetros del modelo en caso necesario.

El crecimiento de \bar{x}_t y la consiguiente existencia de un espacio de dinámica múltiple son características también propias de las tareas de seguimiento de múltiples objetos, tal y como se explica en el capítulo 2. Debido a ello, el uso de un paso MCMC aplicado al problema de MTT aparece en varias referencias, como en [Khan05] analizado en dicho capítulo.

En el algoritmo “Selección y Movimiento” de Berzuini la etapa de “movimiento” propuesto se generaliza de modo que el instante temporal de su implementación, las partículas y/o los elementos del vector de estado implicados en él, e incluso el valor del núcleo $K(\bar{x}_{0:t}, \bar{x}_{0:t})$ son

variables. De este modo será posible asignar una función de transición en una iteración concreta a un parámetro concreto del vector de estado $K_1(\bar{x}_{1,0:t1}, \hat{\bar{x}}_{1,0:t1})$, utilizar otra diferente para otro parámetro $K_2(\bar{x}_{2,0:t2}, \hat{\bar{x}}_{2,0:t2})$ o no asignar ninguna a alguno de ellos en la iteración siguiente.

Del párrafo anterior se concluye que el algoritmo descrito en por Berzuini tiene como principal ventaja frente al paso MCMC estándar que la modificación de las partículas de S_t se realiza de forma completamente controlada.

En esta tesis, se plantea también una etapa de transformación de la creencia $p(\bar{x}_t | \bar{y}_{1:t})$ generada por el PF básico, consistente en hacer un movimiento controlado de las partículas de S_t hacia zonas de alta probabilidad de la o las hipótesis de tracking. Para ello se emplea información de la verosimilitud $p(\bar{y}_t | \bar{x}_t)$ y, más concretamente, del proceso de clasificación de medidas $Y_t \Rightarrow G_{1:k,t|in}$ propuesto, tal y como se explica en capítulos posteriores.

El movimiento de partículas S_t que se incluye en la propuesta de estimación aquí planteada es, por tanto, controlado, e implica un tiempo de cómputo mucho menor que el de las alternativas expuestas en este punto, que permite que la aplicación se ejecute en tiempo real.

c) “Muestreo segmentado”:

En [MacCormick00] se presenta otra solución que evita que $S_t' = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ se vea empobrecido en el paso de corrección del algoritmo “*Bootstrap*”, debido a que la verosimilitud $p(\bar{y}_t | \bar{x}_t)$ y la creencia a priori $p(\bar{x}_t | \bar{y}_{1:t-1})$ muestren una relación como la que aparece en la Figura 4.3. La idea consiste en dividir $p(\bar{y}_t | \bar{x}_t)$ en varias y usarlas de forma individual, para obtener el vector de pesos $\bar{w}_t \rightarrow w(\bar{x}_{0:t}) = w(\bar{x}_{0:t-1}) \cdot p(\bar{y}_t | \bar{x}_t)$ por etapas. El método, ya expuesto en el capítulo 2, toma el nombre de “*Muestreo Segmentado*” (“*Partitioned Sampling*”).

Evidentemente, esta solución solo es aplicable a casos de estimación en los que la verosimilitud puede descomponerse en otras PDFs que no produzcan el problema de empobrecimiento descrito. Este hecho se cumple en el caso de que algunas de las componentes de \bar{x}_t sean independientes de otras, tal y como se muestra en la expresión siguiente:

$$p(\bar{y}_t | \bar{x}_t) = p(\bar{y}_t | \bar{x}_{1,t}) + p(\bar{y}_t | \bar{x}_{2,t}) / \bar{x}_t = \{\bar{x}_{t,1}, \bar{x}_{t,2}\} \quad <4.33>$$

En [Doucet01] Oudjane propone un proceso muy similar al del “*Partitioned Sampling*”, denominado “*Progressive Correction*”, consistente en dividir el paso de corrección del PF en dos etapas. Tanto el objetivo como el procedimiento de este proceso son idénticos a los de la propuesta de MacCormick: dividir $p(\bar{y}_t | \bar{x}_t)$ en dos distribuciones, de modo que sea posible controlar en mayor medida en el proceso de obtención de pesos \bar{w}_t del algoritmo “*Bootstrap*”:

$$w(\bar{x}_{0:t}) = w(\bar{x}_{0:t-1}) \cdot p(\bar{y}_t | \bar{x}_t) = w(\bar{x}_{0:t-1}) \cdot p(\bar{y}_t | \bar{x}_{1,t}) + w(\bar{x}_{0:t-1}) \cdot p(\bar{y}_t | \bar{x}_{2,t}) \quad <4.34>$$

La técnica de dividir las PDFs implicadas en el PF en los distintos procesos de cálculo envueltos en éste es utilizada por otros autores en las diferentes alternativas propuestas a la versión básica del filtro bayesiano, como algunas de las expuestas en el apéndice I. En general, el fin perseguido con el empleo de estas técnicas es aumentar el control sobre la evolución de S_t , e incrementar así la robustez del filtro final.

d) “Auxiliary Particle Filter” (APF):

En [Pitt99] se propone el uso de una nueva función de aproximación a la creencia $q(\bar{x}_{0:t} | \bar{y}_{1:t}) = q(\bar{x}_t, \xi | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ en el algoritmo básico SIR, que incorpora información sobre la verosimilitud $p(\bar{y}_t | \bar{x}_t)$, que no aparece en la versión “*Bootstrap*” del PF (en el que se recuerda que $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t}) = p(\bar{x}_t | \bar{x}_{t-1})$). El algoritmo resultante fue denominado por sus autores “*Filtro de Partículas Auxiliar*” (“*Auxiliary Particle Filter*”, APF o también “*SIR Auxiliar*”, ASIR “*Auxiliary SIR*”), y su formalización matemática consiste en definir la función de aproximación a la creencia, más parecida a la óptima, de la siguiente forma:

$$q(\bar{x}_t, \xi | \bar{x}_{0:t-1}, \bar{y}_{1:t}) \propto p(\bar{x}_{0:t-1}^{(1:n)} | \bar{y}_{1:t-1}) \cdot p(\bar{x}_t | \bar{x}_{t-1}) \cdot p(\bar{y}_t | \xi_t^{(1:n)}), \quad <4.35>$$

donde ξ es un parámetro estadístico de la PDF $p(\bar{x}_t | \bar{x}_{t-1})$ que caracteriza al modelo de estado del sistema implicado, elegido según de la aplicación.

Con esta nueva forma de la función de aproximación, la ecuación <4.5>, que define la forma de cálculo de los pesos del set de partículas \bar{w}_t , se modifica de la siguiente forma:

$$\begin{aligned} w(\bar{x}_{0:t}) &= \frac{p(\bar{y}_{1:t} | \bar{x}_{0:t}) \cdot p(\bar{x}_{0:t})}{q(\bar{x}_t, \xi | \bar{x}_{0:t-1}, \bar{y}_{1:t})} \Bigg|_{i=1:n} = \frac{p(\bar{x}_{0:t} | \bar{y}_{1:t})}{p(\bar{x}_{0:t-1}^{(i)} | \bar{y}_{1:t-1}) \cdot p(\bar{x}_t | \bar{x}_{t-1}) \cdot p(\bar{y}_t | \xi_t^{(i)})} \Bigg|_{i=1:n} = \\ &= \frac{p(\bar{x}_{0:t-1} | \bar{y}_{1:t-1}) \cdot p(\bar{x}_t | \bar{x}_{t-1}) \cdot p(\bar{y}_t | \bar{x}_t)}{p(\bar{x}_{0:t-1}^{(i)} | \bar{y}_{1:t-1}) \cdot p(\bar{x}_t | \bar{x}_{t-1}) \cdot p(\bar{y}_t | \xi_t^{(i)})} \Bigg|_{i=1:n} = \frac{p(\bar{y}_t | \bar{x}_t)}{p(\bar{y}_t | \xi_t^{(i)})} \Bigg|_{i=1:n} \end{aligned} \quad <4.36>$$

Tal y como se puede apreciar en la expresión anterior, la modificación introducida en [Pitt99] al PF básico conlleva la incorporación en $\bar{w}_t \rightarrow w(\bar{x}_{0:t})$ de información sobre la relación existente entre Y_t y el modelo de actuación $p(\bar{x}_t | \bar{x}_{t-1})$, a través de $p(\bar{y}_t | \xi_t^{(1:n)})$. De este modo se evita el empobrecimiento del set cuando $p(\bar{y}_t | \bar{x}_t)$ es muy diferente a la PDF a priori $p(\bar{x}_t | \bar{y}_{1:t-1})$ (ver la Figura 4.3).

Sin embargo ([Merwe01]) el ASIR no resuelve el problema de empobrecimiento en caso de que la verosimilitud coincida en mayor medida con la probabilidad a priori, por lo que este algoritmo no resuelve el problema de forma global.

Una propuesta muy semejante a la descrita en este apartado es presentada en [Doucet01] por Godsill y Clapp. Estos autores plantean usar como función de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ una PDF mezcla entre $p(\bar{y}_t | \bar{x}_t)$ y la densidad que caracteriza al modelo de

actuación $p(\bar{x}_t | \bar{x}_{t-1})$, a la cual denominan “*Bridging Density*” (“*Densidad Puente*”). El uso de esta densidad combinada resuelve también en este caso el problema de empobrecimiento del algoritmo “*Bootstrap*”, si bien la carga computacional que conlleva el cálculo de la “*Densidad Puente*” imposibilita su uso en aplicación con restricciones de tiempo real, como la de interés.

e) “*Importance Sampling Condensation*” (“*ICondensation*”):

Todas las alternativas de generación de una función de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ analizadas hasta el momento son directas, pues se basan en la modificación de la versión “*Bootstrap*” del PF. Existen en la literatura científica también soluciones indirectas, basadas en el uso de un estimador complementario para obtener el valor más adecuado para esta PDF. En [Isard98b] se plantea por primera vez una de estas técnicas indirectas para generar la densidad $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$, dando lugar al algoritmo “*Importance Sampling Condensation*”, conocido como “*ICondensation*”, y ya presentado en el capítulo 2.

En el algoritmo “*ICondensation*”, que en [Isard98b] se aplica al tracking de objetos en imágenes, se utiliza un filtro complementario al PF para realizar la detección y el seguimiento de blobs de color en la imagen, que permite:

1. Generar un valor de $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ más semejante a la verosimilitud $p(\bar{y}_t | \bar{x}_t)$ de los targets a seguir, a usar en un PF que realiza el seguimiento fino de la posición estos en base a un set de datos independientes de los implicados en el modelo de observación (medidas de contorno, en [Isard98b]).
2. Detectar a grandes rasgos nuevos objetos en la escena de forma rápida y fácil.

En la aplicación de interés de la tesis, la segunda de las funcionalidades enumeradas es tan importante o más que la primera, pues permite la utilización del sistema de tracking en situaciones en las que el número de targets a seguir en la escena $k_{out,t}$ es variable. Sin embargo, esta funcionalidad no es aprovechada en [Isard98b], tal y como se explica ya en el capítulo 2.

El proceso de MTT en este algoritmo se lleva a cabo, por tanto, en dos estimadores. El filtro maestro trabaja sobre las características de contorno del objeto a seguir, para estimar el valor de $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ que ha de usarse en el filtro esclavo, que por su parte, trabaja sobre las características de color del target.

El proceso de estimación auxiliar incluido en la secuencia básica del PF del algoritmo “*ICondensation*” entra en funcionamiento en el paso de corrección, cuando la diferencia entre la densidad de transición $p(\bar{x}_t | \bar{x}_{t-1})$ y la de verosimilitud supera un umbral predeterminado, que puede producir los problemas de empobrecimiento en la estimación. En ese caso se utiliza la propuesta de $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ que arroja el PF complementario.

El uso de un segundo PF para proporcionar el valor de la densidad de aproximación $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$, tiene el inconveniente de que este filtro necesita también una propuesta de

$q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ que asegure la exactitud de la PDF generada, y por ende, la robustez del filtro maestro, o lo que es lo mismo en la aplicación de interés, del sistema de tracking.

El algoritmo “*ICondensation*” permite realizar la estimación de múltiples objetos mediante el método del vector de estado aumentado χ_t . Con una única densidad de creencia conjunta $p(\chi_t | \bar{y}_{1:t})$, este filtro permite realizar el seguimiento de cada uno de los targets en la escena, y además, del número de ellos $k_{out,t}$ en la misma. Como ya se comenta en el capítulo 2, esta técnica de estimación múltiple tiene como mayor inconveniente la difícil gestión computacional de un vector de estado cuyas dimensiones son dinámicas.

Este algoritmo ha sido extensamente utilizado de forma directa o modificada en trabajos de seguimiento basado en imágenes ([Chen02], [Torma03] o [Chang06]), si bien en esta tesis se propone un método más general y eficiente para realizar la tarea de MTT propuesta.

f) “*Extended Kalman Particle Filter*” (EKPF):

Otro de los métodos más utilizados por la comunidad científica para obtener de forma indirecta un valor próximo al óptimo de la función de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ es el de estimarlo mediante un filtro de Kalman ([Pitt99], [Merwe01], [Doucet01], [Arulampalam02] o [Ristic04]).

El objetivo del algoritmo, llamado generalmente “*Filtro de Partículas con Filtro de Kalman Extendido*” (“*Extended Kalman Particle Filter*”, EKPF), es el mismo que el perseguido con el ASIR: diseñar una densidad $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ que incluya información acerca no solo del modelo de estado $p(\bar{x}_t | \bar{x}_{t-1})$, sino también del de observación del sistema implicado, a través de la verosimilitud $p(\bar{y}_t | \bar{x}_t)$. Sin embargo, en el caso del EKPF se utiliza un estimador externo (un EKF) para obtener el valor de esta PDF $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ (técnica indirecta), en lugar de inyectar información acerca del conjunto Y_t en la función $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ (técnica directa del ASIR).

La propuesta de $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ del EKPF es individual para cada partícula, y de valor: $p(\hat{\bar{x}}_t^{(i)} | \bar{x}_{0:t-1}^{(i)}, \bar{y}_{1:t})$.

En el EKPF se establece que el valor de las partículas $S_{t|t-1} = \{\bar{s}_{i,t|t-1} = \bar{x}_{t|t-1}^{(i)} / i = 1:n\}$ ha de ser actualizado tras la etapa de predicción del PF con las fórmulas recursivas del KF (presentadas en el capítulo 2) obteniendo así un nuevo valor para éstas $\hat{S}_t = \{\hat{\bar{s}}_{i,t} = \hat{\bar{x}}_t^{(i)} / i = 1:n\}$. Para ello es necesario linealizar, mediante una serie de Taylor, tanto el modelo de actuación ($\bar{x}_{t|t-1}^{(1:n)} = f(\bar{x}_{t-1}^{(1:n)}, \bar{u}_{t-1}, \bar{v}_{t-1}) = A \cdot \bar{x}_{t-1}^{(1:n)} + B \cdot \bar{u}_{t-1} + \bar{v}_{t-1}$) como el de observación ($\bar{y}_t^{(1:n)} = f(\bar{x}_t^{(1:n)}, \bar{u}_t, \bar{o}_t) = C \cdot \bar{x}_t^{(1:n)} + D \cdot \bar{u}_t + \bar{o}_t$) del sistema implicado en la estimación.

A partir del valor así actualizado de las partículas $\widehat{S}_t = \{\widehat{s}_{i,t} = \widehat{x}_t^{(i)} / i = 1:n\}$ se calcula la función de aproximación a la creencia que caracteriza a cada partícula $p(\widehat{x}_t^{(i)} | \widehat{x}_{0:t-1}^{(i)}, \widehat{y}_{1:t})$, que en esta versión del PF, debido al uso de un EKF para el cálculo de \widehat{S}_t , es una densidad gaussiana de media $\widehat{x}_t^{(i)}$ y varianza $P_t^{(i)}$ para cada partícula $\widehat{x}_t^{(i)}$, es decir:

$$p(\widehat{x}_t^{(i)} | \widehat{x}_{0:t-1}^{(i)}, \widehat{y}_{1:t}) = \text{norm}(\widehat{x}_t^{(i)}, P_t^{(i)}) \quad <4.37>$$

Así, el paso de corrección del PF se transforma en uno de “*actualización + muestreo*”, puesto que en este paso se obtiene, para cada partícula $\widehat{x}_{t-1}^{(i)}$, la función $p(\widehat{x}_t^{(i)} | \widehat{x}_{0:t-1}^{(i)}, \widehat{y}_{1:t})$ mostrada en <4.37> y, mediante el muestreo de ésta, un nuevo valor de la partícula $\widehat{x}_t^{(i)}$.

De este modo, el valor de la PDF $p(\widehat{x}_t^{(i)} | \widehat{x}_{0:t-1}^{(i)}, \widehat{y}_{1:t})$ se utiliza en esta propuesta para obtener el conjunto $\widehat{S}_t = \{\widehat{s}_{i,t} = \widehat{x}_t^{(i)} / i = 1:n\}$ que representan a la creencia a la salida de la corrección.

Además, la misma PDF de aproximación a la creencia $p(\widehat{x}_t^{(i)} | \widehat{x}_{0:t-1}^{(i)}, \widehat{y}_{1:t})$ obtenida con el EKF se utiliza con el objetivo primeramente planteado en este apartado: la obtención del peso asociado a cada partícula $w_t^{(i)}$, que se calcula utilizando la $p(\widehat{x}_t^{(i)} | \widehat{x}_{0:t-1}^{(i)}, \widehat{y}_{1:t})$ gaussiana característica de cada una de ellas:

$$w(\widehat{x}_{0:t}^{(i)}) = w_t^{(i)} = w_{t-1}^{(i)} \cdot \frac{p(\widehat{y}_t | \widehat{x}_t) \cdot p(\widehat{x}_t | \widehat{x}_{t-1})}{p(\widehat{x}_t^{(i)} | \widehat{x}_{0:t-1}^{(i)}, \widehat{y}_{1:t})} \quad <4.38>$$

La técnica consistente en calcular una función continua $p(\widehat{x}_t^{(i)} | \widehat{x}_{0:t-1}^{(i)}, \widehat{y}_{1:t})$ que se aproxime a la creencia $p(\widehat{x}_t | \widehat{y}_{1:t})$ para obtener, mediante su muestreo, el nuevo conjunto S_t que la caracterice es también muy habitual en las soluciones al problema de empobrecimiento del set propuestas por la comunidad científica, tal y como se expone en el apéndice I, donde se describen otras alternativas del PF muy empleadas en el área de interés. El mayor inconveniente de esta técnica es, de nuevo, la carga computacional que conlleva el cálculo de una función continua para cada partícula, aunque sea parametrizada, como la expuesta en este punto ($p(\widehat{x}_t^{(i)} | \widehat{x}_{0:t-1}^{(i)}, \widehat{y}_{1:t}) = \text{norm}(\widehat{x}_t^{(i)}, P_t^{(i)})$).

Evidentemente, la linealización comentada a la que se somete el modelo no lineal del sistema para poder obtener una mejor función de aproximación a la creencia $p(\widehat{x}_t^{(i)} | \widehat{x}_{0:t-1}^{(i)}, \widehat{y}_{1:t})$ disminuye la exactitud de la estimación. Por esta razón el EKPF solo es eficiente en aplicaciones en las que el modelo del sistema involucrado no sea excesivamente no lineal.

Como se comenta en el capítulo 2, utilizando el algoritmo UKF ([Julier97]) en lugar del EKF, es posible obtener una versión lineal más semejante al real, del modelo no lineal del sistema implicado en el proceso de estimación. El empleo del UKF en lugar del EKF a la idea

presentada en este punto da lugar al algoritmo UPF “*Unscented Particle Filter*” ([Merwe01]), que, debido a su gran aceptación por la comunidad científica en los últimos años en distintos campos, se explican en mayor detalle en el apéndice I.

En el algoritmo UPF se propone, por tanto básicamente, el uso de una densidad de aproximación a la creencia de valor $p(\hat{\vec{x}}_t^{(i)} | \vec{x}_{0:t-1}^{(i)}, \vec{y}_{1:t})$ muy parecido a su valor óptimo $p(\vec{x}_t | \vec{x}_{0:t-1}, \vec{y}_{1:t})$, de modo que se minimizan los inconvenientes de inexactitud inherentes en el EKPF, debidos a la linealización del modelo del sistema mediante series de Taylor.

El alto coste computacional del UKF implica que el cálculo en tiempo real de la función $p(\hat{\vec{x}}_t^{(i)} | \vec{x}_{0:t-1}^{(i)}, \vec{y}_{1:t})$ es incluso más difícil que en el EKPF ([Rui01]), por lo que el uso de cualquiera de estos algoritmos no es aconsejable desde el punto de vista de los objetivos planteados en esta tesis.

Merece la pena comentar en este punto que en algunas publicaciones (p. ej. [Wright03]) se denomina al EKPF simplemente “*Filtro de Partículas Extendido*” (“*eXtended Particle Filter*”, XPF). Tal y como se expone en el capítulo 3, este nombre se usa en este documento para referenciar a una versión del PF que se extiende para poderse utilizar como estimador de múltiples sistemas, y que no tiene, por tanto, nada que ver con el EKPF. El nombre de “*Filtro de Partículas Extendido*” se restringe, por tanto, en este documento a la versión del PF explicada en el punto siguiente para evitar confusión.

4.3. “*Filtro de Partículas Extendido*” (XPF)

Una vez analizado el funcionamiento del PF básico y de las versiones del mismo más usadas por la comunidad científica para resolver problemas de estimación como el de interés, en este apartado se describe el “*Filtro de Partículas Extendido*” (XPF, “*eXtended Particle Filter*”). Este algoritmo es, como se explica en el capítulo anterior, la versión del PF base de la propuesta de estimación multimodal presentada en esta tesis: el “*Filtro de Partículas Extendido con Proceso de Clasificación*” (XPFCP, “*eXtended Particle Filter with Clustering Process*”).

El XPF fue propuesto en [Koller-Meier01], analizado en el capítulo 2, pero debido a su baja robustez no se encuentra en referencias posteriores. En esta sección se describe la funcionalidad de sus distintas etapas y se resaltan sus puntos débiles, que constituyen la razón de su posterior desuso. A partir de esta descripción, en los párrafos siguientes se introducen también las mejoras a la funcionalidad del XPF que dan lugar al XPFCP propuesto finalmente, que se describen en detalle en el capítulo 6.

La funcionalidad del XPF se obtiene, a su vez, incluyendo varias modificaciones al flujograma del algoritmo “*Bootstrap*”. El fin de éstas es la adaptación del filtro a tareas de estimación multimodal, de modo que la creencia a su salida $p(\vec{x}_t | \vec{y}_{1:t})$ permita modelar el comportamiento de un número cambiante de sistemas $k_{out,t}$, objetos en movimiento en la aplicación de MTT de interés.

El carácter multimodal del XPF es la razón fundamental del uso de este algoritmo como base del XPFCP diseñado en la tesis. Tal y como se extrae del análisis expuesto en el capítulo 2, el XPF es una de las pocas alternativas presentadas por la comunidad científica que cumple la especificación de multimodalidad, y su consecución es esencial en los objetivos perseguidos.

Por otro lado, el hecho de que el flujograma del XPF esté basado en el del algoritmo “*Bootstrap*” es una razón añadida para su elección como base del XPFCP. La baja carga computacional y la gran sencillez funcional del algoritmo “*Bootstrap*”, y por ende del XPF, facilitan el diseño de las modificaciones que, aplicadas a estos algoritmos, permiten conseguir el resto de las especificaciones perseguidas para el XPFCP (robustez, fiabilidad, flexibilidad y eficiencia) sin detrimento de su ejecución en tiempo real. En contrapartida, las modificaciones que se proponen han de resolver los problemas comentados de empobrecimiento característicos del algoritmo “*Bootstrap*” y de robustez propios del XPF.

4.3.1. Funcionalidad del XPF

La Figura 4.5 presenta el diagrama de funcionamiento del XPF, cuyas etapas se describen a continuación de forma detallada. En la figura, los pasos del XPF que han sido modificados respecto a su definición en el PF básico (en la Figura 4.1) se muestran sobre fondo azul.

Por otro lado, la Figura 4.6 expone un ejemplo de funcionamiento del XPF, en el que se analiza la evolución de $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ a lo largo de las distintas etapas del algoritmo, para $\bar{x} = x$ unidimensional³. Esta figura es equivalente a la Figura 4.2, en la que se muestra esta misma evolución para el caso de un PF básico, por lo que las diferencias entre ambas figuras presentan de forma clara la diferente funcionalidad de ambos algoritmos.

a) Paso de “*re-inicialización*”

Para conseguir que la creencia $p(\bar{x}_t | \bar{y}_{1:t})$, y por ende el proceso de estimación, se adapte dinámicamente al número variable de hipótesis de seguimiento que, en la aplicación de tracking de interés, aparecen en el entorno en cada momento, es necesario, tal y como se describe en el capítulo 3, introducir un paso de “*re-inicialización*” de la creencia en el lazo básico de ejecución del PF (ver la Figura 4.5).

Esta nueva funcionalidad, que es además la más característica del XPF presentado en [Koller-Meier01], se conserva en el XPFCP propuesto en la tesis para resolver tarea de MTT. De esta forma se modifica la creencia $p(\bar{x}_t | \bar{y}_{1:t})$ a la salida del filtro, incorporando en ésta picos de probabilidad que representan a todos los objetos de la escena: tanto los existentes ($n - n_m = 10$ partículas azules en la Figura 4.6) como los que no lo eran en la iteración anterior del filtro ($n_m = 4$ partículas verdes en la figura).

³ En este ejemplo $\gamma = \frac{n_m}{n} = \frac{4}{14}$.

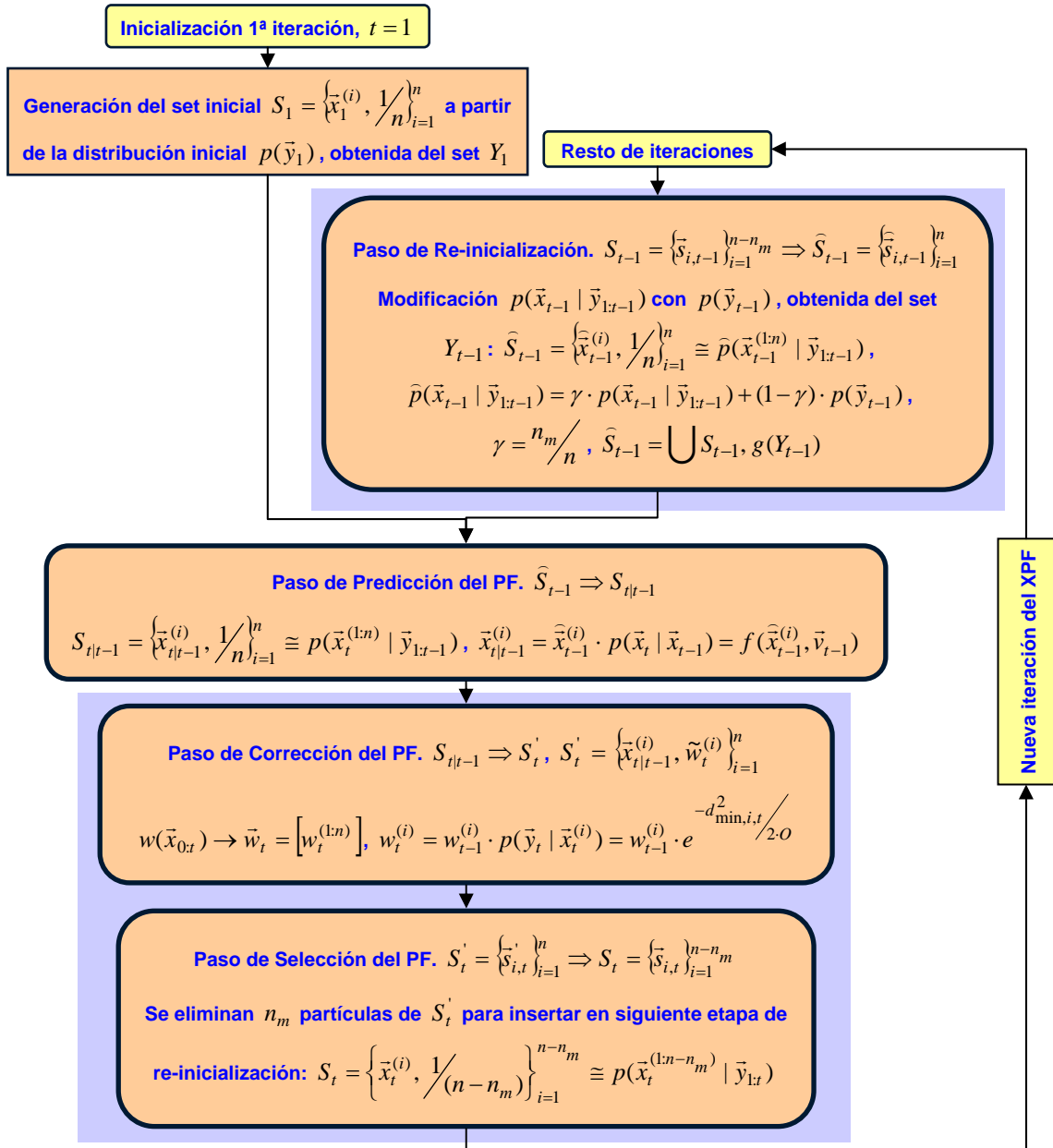


Figura 4.5. Diagrama funcional del algoritmo XPF. Los pasos modificados respecto a la versión básica del PF de la Figura 4.1 se muestran sobre fondo azul.

Este es el único método encontrado en la literatura científica que permite usar un único PF basado en vector de estado simple (no extendido) para realizar la tarea de estimación de un número variable de sistemas $k_{out,t}$.

De esta forma se modifica la densidad $p(\bar{x}_{t-1} | \bar{y}_{1:t-1})$ sobre la que se ejecuta la etapa de predicción del PF básico, que pasa a declararse como $\widehat{p}(\bar{x}_{t-1} | \bar{y}_{1:t-1})$ (ver la Figura 4.5).

Tal y como se explica en el capítulo 3 y se observa en la Figura 4.6, la forma de desarrollar la funcionalidad de este paso consiste en insertar n_m nuevas partículas en el set

$S_{t-1} = \left\{ \bar{x}_{t-1}^{(i)}, \frac{1}{n-n_m} \right\}_{i=1}^{n-n_m}$, de modo que el set de salida de la etapa de re-inicialización $\hat{S}_{t-1} = \bigcup S_{t-1}, g(Y_{t-1}) = \left\{ \bar{x}_{t-1}^{(i)}, \frac{1}{n} \right\}_{i=1}^n$ cuente de nuevo con el número total de éstas n , establecido como parámetro del filtro.

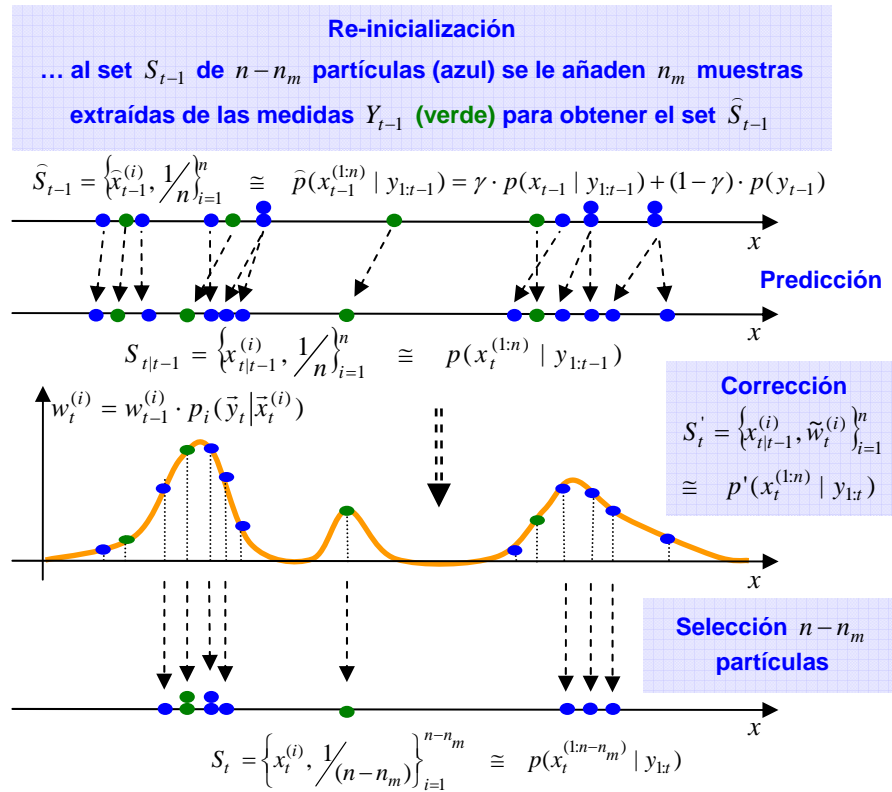


Figura 4.6. Representación unidimensional de la evolución de $S_t = \left\{ \bar{x}_t^{(i)}, \tilde{w}_t^{(i)} \right\}_{i=1}^n$ en las diferentes etapas del XPF. Los pasos modificados respecto a la versión básica del PF de la Figura 4.2 se indican con fondo azul. Las partículas nuevas insertadas en el paso de re-inicialización $\hat{S}_{i,t-1} / i = n-n_m : n$ se muestran en color verde.

En [Koller-Meier01] se especifica que las n_m partículas insertadas en este paso ($g(Y_{t-1}) = \left\{ \bar{s}_{i,t-1} \right\}_{i=n-n_m}^n = \left\{ \bar{x}_{t-1}^{(i)}, \frac{1}{n} \right\}_{i=n-n_m}^n$) se obtienen del conjunto de medidas $Y_{t-1} = \left\{ \bar{y}_{t-1}^{(j)} \right\}_{j=1}^{m_{t-1}}$, o lo que es equivalente, de la densidad $p(\bar{y}_{t-1})$. Debido a ello, la primera iteración del XPF solo puede ejecutarse una vez que se cuenta con el primer set de medidas Y_1 , razón por la cuál en el flujograma de la Figura 4.5 se indica que esta primera iteración se corresponde con el instante $t=1$, en lugar de $t=0$, como ocurre con el PF básico (Figura 4.1).

En el capítulo 2 se expone el modelo usado en la tarea de estimación, basada en el PF, e implicado en la aplicación de seguimiento multi-objeto interés. Se trata de un modelo de velocidad constante (CV), cuya definición matemática, ya expuesta en las expresiones <2.6> y <2.8> de dicho capítulo, se repiten también a continuación por claridad:

$$\bar{\mathbf{x}}_t = \begin{bmatrix} x_t \\ y_t \\ z_t \\ vx_t \\ vz_t \end{bmatrix} = f(\bar{\mathbf{x}}_{t-1}, \bar{\mathbf{v}}_{t-1}) = A \cdot \bar{\mathbf{x}}_{t-1} + \bar{\mathbf{v}}_{t-1} = \begin{bmatrix} 1 & 0 & 0 & t_s & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & t_s \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \\ vx_{t-1} \\ vz_{t-1} \end{bmatrix} + \begin{bmatrix} v_{x,t-1} \\ v_{y,t-1} \\ v_{z,t-1} \\ v_{vx,t-1} \\ v_{vz,t-1} \end{bmatrix} \quad <4.39>$$

$$\bar{\mathbf{y}}_t = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = h(\bar{\mathbf{x}}_t, \bar{\mathbf{o}}_t) = C \cdot \bar{\mathbf{x}}_t + \bar{\mathbf{o}}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ z_t \\ vx_t \\ vz_t \end{bmatrix} + \begin{bmatrix} o_{x,t} \\ o_{y,t} \\ o_{z,t} \end{bmatrix} \quad <4.40>$$

Gracias a la sencillez del modelo expuesto, el uso de algunas de las medidas de set $Y_{t-1} = \{\bar{\mathbf{y}}_{t-1}^{(j)}\}_{j=1}^{m_{t-1}}$ como nuevas partículas a insertar en el paso de re-inicialización $g(Y_{t-1}) = \{\bar{\mathbf{x}}_{t-1}, \frac{1}{n}\}_{i=n-n_m}^n$, sólo implica añadir a sus componentes $\bar{\mathbf{y}}_{t-1} = [x_{t-1} \ y_{t-1} \ z_{t-1}]^T$ las de velocidad $[vx_{t-1} \ vz_{t-1}]^T$, que completan el contenido del vector de estado representado por cada partícula $\bar{\mathbf{x}}_{t-1} = [x_{t-1} \ y_{t-1} \ z_{t-1} \ vx_{t-1} \ vz_{t-1}]^T$. En el XPF, las componentes de velocidad son inicializadas con valor nulo en cada partícula. Este valor evoluciona a lo largo de las iteraciones del filtro, hasta ajustarse a la velocidad del objeto al que representa cada partícula, gracias al ruido $\bar{\mathbf{v}}_{t-1}$ característico del modelo CV y al propio funcionamiento en tres etapas (predicción, corrección y selección) del estimador bayesiano.

La incorporación de este paso de re-inicialización, permite que la creencia a priori $p(\bar{\mathbf{x}}_t | \bar{\mathbf{y}}_{1:t-1})$ que se obtiene tras el de predicción del PF incluya la información de verosimilitud $p(\bar{\mathbf{y}}_{t-1} | \bar{\mathbf{x}}_{t-1})$ ausente en la función de aproximación a la creencia $q(\bar{\mathbf{x}}_t | \bar{\mathbf{x}}_{0:t-1}, \bar{\mathbf{y}}_{1:t}) = p(\bar{\mathbf{x}}_t | \bar{\mathbf{x}}_{t-1})$ propia del filtro “*Bootstrap*”.

Este nuevo paso es, por tanto, una solución alternativa a las anteriormente expuestas (paso MCMC, de “*Edición Previa*” o de “*Rechazo Selectivo*”, o “*Muestreo Segmentado*”) para resolver el problema de empobrecimiento propio del algoritmo “*Bootstrap*”, pues tiene el mismo efecto de mover las partículas a zonas del espacio de interés más próximas a la verosimilitud, evitando el problema que se expone en la anterior Figura 4.3. Además, la etapa de re-inicialización propuesta aporta como ventaja frente a los algoritmos mencionados, la adaptación de la creencia $p(\bar{\mathbf{x}}_t | \bar{\mathbf{y}}_{1:t})$ al modelado de un número variable de hipótesis de estimación $k_{out,t}$, especificación fundamental de la tesis.

Desde un punto de vista matemático, la etapa de re-inicialización conlleva la modificación de la PDF de entrada al XPF ($p(\bar{\mathbf{x}}_{t-1} | \bar{\mathbf{y}}_{1:t-1}) \Rightarrow \bar{p}(\bar{\mathbf{x}}_{t-1} | \bar{\mathbf{y}}_{1:t-1})$) que se muestra a continuación, y ya fue expuesta en el capítulo anterior:

$$\bar{p}(\bar{\mathbf{x}}_{t-1} | \bar{\mathbf{y}}_{1:t-1}) = \gamma \cdot p(\bar{\mathbf{x}}_{t-1} | \bar{\mathbf{y}}_{1:t-1}) + (1 - \gamma) \cdot p(\bar{\mathbf{y}}_{t-1}) \quad <4.41>$$

$$\gamma = n_m / n, \quad <4.42>$$

donde el parámetro γ representa el porcentaje de partículas que se insertan en este paso de re-inicialización n_m , frente al número total de éstas n . En el ejemplo de evolución del XPF, mostrado en la Figura 4.6, $\gamma = 4/14$.

Es necesario tener en cuenta que el valor de γ ha de ser suficientemente bajo de modo que no modifiquen el rigor teórico de la evolución de la creencia $p(\bar{x}_t | \bar{y}_{1:t})$ en el PF, y se mantenga la funcionalidad de la etapa de re-inicialización del XPF.

b) Paso de predicción

El paso de corrección es en el XPF idéntico al del algoritmo “*Bootstrap*”. En esta etapa del XPF las partículas procedentes del paso de re-inicialización $\hat{S}_{t-1} = \{\hat{\bar{x}}_{t-1}^{(i)}, 1/n\}_{i=1}^n \cong \hat{p}(\bar{x}_{t-1}^{(1:n)} | \bar{y}_{1:t-1})$, se propagan con el modelo de estado del sistema $p(\bar{x}_t | \bar{x}_{t-1}) \cong f(\bar{x}_{t-1}^{(1:n)}, \bar{v}_{t-1})$ elegido para el proceso de estimación (ver la Figura 4.5). De esta forma se obtiene la forma discretizada del valor a priori de la creencia $S_{t|t-1} = \{\bar{x}_{t|t-1}^{(i)}, 1/n\}_{i=1}^n \cong p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t-1})$ a través de la expresión:

$$\bar{x}_{t|t-1}^{(1:n)} = \hat{\bar{x}}_{t-1}^{(1:n)} \cdot p(\bar{x}_t | \bar{x}_{t-1}) = f(\hat{\bar{x}}_{t-1}^{(1:n)}, \bar{v}_{t-1}) \quad <4.43>$$

Tal y como se expone en el punto anterior, el modelo de actuación usado en la aplicación de interés ($p(\bar{x}_t | \bar{x}_{t-1}) \cong f(\bar{x}_{t-1}^{(1:n)}, \bar{v}_{t-1})$) es uno de velocidad constante (CV) caracterizado por las expresiones <4.39> y <4.40> anteriores.

Este modelo es lineal y muy sencillo, y como se explica en el capítulo 2, es el más usado por la comunidad científica en la tarea de estimación implícita en problemas de tracking. El PF y el XPF son filtro bayesianos que no plantean ninguna restricción al tipo de modelo del sistema implicado en el filtrado, por lo que el elegido podría incorporar cualquier elemento no lineal, sin que ello tuviese ninguna repercusión en la fiabilidad de la estimación. Desde este punto de vista, como se comenta en el capítulo anterior, el XPF aporta a la propuesta de estimación la flexibilidad especificada en los objetivos perseguidos.

En el punto anterior se muestra que el vector de estado $\bar{x}_t = [x_t \quad y_t \quad z_t \quad vx_t \quad vz_t]^T$, usado en el proceso de estimación basado en el PF para resolver el problema de MTT, incluye las componentes de velocidad del movimiento del objeto al que representa cada partícula $S_{t-1} = \{\hat{\bar{x}}_{t-1}^{(i)}, 1/n\}_{i=1}^n$. Gracias a esta caracterización, cada partícula del filtro evoluciona con el modelo antes descrito ($f(\hat{\bar{x}}_{t-1}^{(1:n)}, \bar{v}_{t-1})$) en función de la velocidad del objeto al que representa en la aplicación de tracking $[vx_{t-1} \quad vz_{t-1}]^T$, sin que para ello sea necesario establecer una asociación explícita entre las partículas y los targets.

Por tanto, el uso en esta tesis de un modelo sencillo e idéntico para todos los targets implicados en la tarea de seguimiento aporta, de nuevo (ver apartado anterior), un importante beneficio para el algoritmo propuesto, pues elimina la necesidad de incluir un proceso de asociación explícita para este paso. Este hecho constituye una gran ventaja pues, como se demuestra en el capítulo 2, la complejidad computacional de este tipo de algoritmos es uno de los problemas más importantes y aún sin solución generalizada entre la comunidad científica, en las tareas de MTT como la de interés.

En contrapartida, el modelo de movimiento de los objetos implicados en la tarea de seguimiento ha de ceñirse al propuesto, lo que lleva de nuevo a la especificación en la tesis de un modelo de estado tan genérico como el mostrado.

c) Paso de corrección

El paso de corrección coincide con el XPF del algoritmo “*Bootstrap*”, con los inconvenientes de funcionalidad que ello conlleva. Tal y como se expone en el punto 4.1.5 del capítulo, este algoritmo se caracteriza por el uso del modelo de actuación del sistema $p(\bar{x}_t | \bar{x}_{t-1})$, como función de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$.

A partir de esta definición, y como muestra en la Figura 4.5, en este paso de corrección se realiza el cálculo de los pesos de las partículas $w_t^{(l:n)}$ con la ecuación <4.28> ya mostrada, que se repite aquí por claridad:

$$w(\bar{x}_{0:t}) = w(\bar{x}_{0:t-1}) \cdot p(\bar{y}_t | \bar{x}_t) \rightarrow w_t^{(l:n)} = w_{t-1}^{(l:n)} \cdot p(\bar{y}_t | \bar{x}_t^{(l:n)}) \quad <4.44>$$

Como verosimilitud $p(\bar{y}_t | \bar{x}_t) \cong p(\bar{y}_t | \bar{x}_t^{(l:n)})$ en [Koller-Meier01] se propone usar en el XPF para cada partícula $S_{t|t-1} = \left\{ \bar{x}_{t|t-1}^{(i)}, \frac{1}{n} \right\}_{i=1}^n$ procedente la etapa de predicción, una gaussiana que tiene de media la proyección de la partícula en el espacio XYZ de medidas definido para la aplicación de MTT de interés $(h(\bar{x}_{t|t-1}^{(i)}, \bar{o}_t))$, y como matriz de covarianza la de este modelo de observación: $p(\bar{y}_t | \bar{x}_t^{(i)}) = \text{norm}(h(\bar{x}_{t|t-1}^{(i)}, \bar{o}_t), O)$.

En la Figura 4.7 se muestra un ejemplo de esta función de verosimilitud $p(\bar{y}_t | \bar{x}_t^{(i)})$ para un espacio de medidas unidimensional, en lugar del tridimensional usado en la tesis.

El valor discreto de cada peso $\tilde{w}_t^{(i)}$ se obtiene buscando el máximo de la gaussiana de entre las realizaciones que se obtienen de ésta para cada medida del set $Y_t = \left\{ \bar{y}_t^{(j)} \right\}_{j=1}^{m_t}$ (marcadas en la Figura 4.7 con un triángulo verde excepto el de la más próxima a la media $h(\bar{x}_{t|t-1}^{(i)}, \bar{o}_t)$ que se muestra en color naranja). Este máximo viene dado, por tanto, por la distancia $d_{\min,i,t}$ entre la media de la gaussiana $p(\bar{y}_t | \bar{x}_t^{(i)}) = \text{norm}(h(\bar{x}_{t|t-1}^{(i)}, \bar{o}_t), O)$ y la medida más próxima a ésta (que se supone extraída del sistema al que caracteriza la partícula en la estimación).

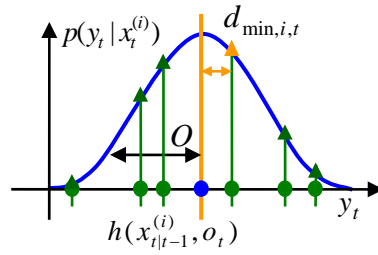


Figura 4.7. Representación unidimensional del proceso de obtención del valor de $p(y_t | x_t^{(i)})$ usado para calcular el peso de $x_{t-1}^{(i)}$, en el paso de corrección del XPF. La línea azul muestra la gaussiana de verosimilitud usada para obtener $\tilde{w}_t^{(i)}$. La media $h(x_{t-1}^{(i)}, o_t)$ (representada con línea naranja) y la varianza O (línea negra) de la función la establece el modelo de observación usado, aplicado al valor concreto de la partícula. En el eje de abscisas se muestran con puntos verdes el conjunto Y_t , cuya aplicación a la gaussiana permite obtener $\tilde{w}_t^{(i)}$.

A partir de la definición descrita, el valor discreto de la verosimilitud $p(\bar{y}_t | \bar{x}_t^{(1:n)})$ mostrada en la ecuación <4.44> anterior, se especifica tal y como se expone a continuación, coincidiendo con lo expuesto en la Figura 4.5. Esta nueva expresión presenta el valor específico del peso $w_t^{(i)}$ que se le asigna a cada partícula $i = 1 : n$ para obtener el set de salida $S_t' = \{\bar{x}_{t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ de la etapa de corrección del XPF:

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot p(\bar{y}_t | \bar{x}_t^{(i)}) = w_{t-1}^{(i)} \cdot e^{-d_{\min,i,t}^2 / 2 \cdot O}, \quad <4.45>$$

posteriormente normalizado a través de la expresión <4.18> repetida aquí por comodidad,

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^n w_t^{(i)}} \quad <4.46>$$

En la expresión <4.45> el valor de distancia $d_{\min,i,t}$ se define de forma general por la ecuación siguiente, a falta de especificar, como ya se ha comentado, el tipo de medida de distancia a usar en este paso:

$$d_{\min,i,t} = \min_{j=1:m_t} \{d(h(\bar{x}_{t-1}^{(i)}, \bar{o}_t), \bar{y}_t^{(j)})\} \quad <4.47>$$

De esta forma, a la partícula $\bar{s}_{i,t|t-1} = \{\bar{x}_{t-1}^{(i)}, 1/n\}$ con una proyección en el espacio de medidas XYZ, $h(\bar{x}_{t-1}^{(i)}, \bar{o}_t)$, semejante a alguna de las m_t medidas de Y_t , se le asigna un peso $\tilde{w}_t^{(i)}$ alto.

En [Koller-Meier01] se propone usar la distancia euclídea en la densidad gaussiana a partir de la que se calculan los pesos de las partículas en esta etapa de corrección. En el capítulo 6 se comenta la conveniencia de emplear esta medida de distancia u otra para esta función. Además, en el apéndice II, se analizan funcionalmente las formas matemáticas más comunes de la medida de distancia en un espacio como el de interés.

Finalmente, merece la pena destacar que la función de cálculo de los pesos $w(\bar{x}_{0:t})$ usada en esta etapa de corrección del XPF es muy semejante a algunas de las alternativas planteadas por la comunidad científica para resolver el problema de empobrecimiento, modificando la función de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ (ver apartado 4.2.2).

En este sentido, un análisis matemático comparativo del valor de esta función $w(\bar{x}_{0:t})$ en el APF (ecuación <4.36>) y en el EKPF (ecuación <4.38>) frente al del XPF (expresión <4.45>) muestra que:

- En los tres casos, el cómputo de la función se realiza de forma específica para cada partícula $w(\bar{x}_{0:t}) \rightarrow w_t^{(1:n)}$. Este hecho simplifica el proceso al no incluir el cálculo del valor continuo de las densidades implicadas ($p(\bar{y}_t | \bar{x}_t)$, $p(\bar{x}_t | \bar{x}_{t-1})$ y $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$), y permite además fortalecer de forma específica el valor de cada uno de los pesos $\bar{w}_t = [\tilde{w}_t^{(1:n)}]$. Además, en el caso de la tarea de MTT de interés en la tesis, el uso de una función $w(\bar{x}_{0:t})$ específica para cada partícula permite establecer de forma implícita la asociación necesaria entre el vector de estado estimado, representado por $\bar{x}_{t|t-1}^{(1:n)}$, y el de medidas representado por Y_t , y desarrollar esta etapa de corrección en función de esta asociación.
- Del mismo modo que ocurre en el caso del APF y en el del EKPF, la PDF básica en el cálculo de los pesos \bar{w}_t en el XPF, $p(\bar{y}_t | \bar{x}_t^{(i)}) = \text{norm}(h(\bar{x}_{t|t-1}^{(i)}, \bar{o}_t), O_t)$, busca la semejanza entre la verosimilitud $p(\bar{y}_t | \bar{x}_t)$ y la creencia a priori $p(\bar{x}_t | \bar{y}_{1:t-1})$, específica del objeto con el que cada partícula $\bar{x}_t^{(i)}$ está implícitamente relacionada en la aplicación de tracking de interés. En el caso del XPF, esta búsqueda está implícita en el uso de la medida del set Y_t más próxima al valor predicho de cada partícula $\bar{x}_{t|t-1}^{(i)}$ en el cálculo de la densidad $p(\bar{y}_t | \bar{x}_t^{(i)})$, siguiendo un esquema de asociación tipo NN. El objetivo en todos los casos es evitar el problema de empobrecimiento del set $S_t' = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$, propio del algoritmo “*Bootstrap*”, y producido por la falta de la similitud entre esas PDFs (ver la Figura 4.3).
- La misma densidad base, analizada en el párrafo anterior, es una gaussiana cuya media se fija en función del valor de la propia partícula, tanto en el caso del XPF $p(\bar{y}_t | \bar{x}_t^{(i)}) = \text{norm}(h(\bar{x}_{t|t-1}^{(i)}, \bar{o}_t), O_t)$ (ver la Figura 4.7) como en el del EKPF $p(\bar{x}_t^{(i)} | \bar{x}_{0:t-1}, \bar{y}_{1:t}) = \text{norm}(\bar{x}_t^{(i)}, P_t^{(i)})$. El uso de una forma gaussiana para esta PDF en lugar de otra más compleja, simplifica también el cálculo de los pesos y facilita, por tanto, la ejecución en tiempo real del filtro.

Por todo ello, la ventaja del uso del sistema de obtención de los pesos propuesto en este punto $w_t^{(1:n)} = w_{t-1}^{(1:n)} \cdot p(\bar{y}_t | \bar{x}_t^{(1:n)})$ frente a los estudiados en el apartado 4.2.2 radica en su bajo coste computacional y en su aportación a la resolución del recurrente problema de

empobrecimiento del set del PF. Además, la propuesta aquí analizada no plantea ninguna restricción de modelo, al contrario que otros algoritmos como el APF y EKPF.

d) Paso de selección

El paso de selección del XPF es idéntico al del algoritmo “*Bootstrap*” con la excepción, ya comentada en el capítulo anterior, de que el proceso de resampling usado en esta etapa se modifica de modo que el número de partículas que constituyen la salida final del estimador bayesiano $S_t = \left\{ \bar{x}_t^{(i)}, \frac{1}{n} \right\}_{i=1}^{n-n_m}$ ha de adaptarse al número de ellas que se espera a la entrada del filtro en su siguiente iteración.

El objetivo de este procedimiento es que el número global de partículas n se mantenga constante a lo largo de las sucesivas iteraciones del XPF, conservándose así su t_{exe} , y facilitándose por tanto la consecución de la especificación de tiempo real.

En algunos trabajos ([Hue02], [Gustafsson02] o [Kreucher05]) se propone ejecutar este paso de re-muestreo de forma selectiva, en función de si el número aproximado de partículas eficaces \hat{n}_{eff} (definido en el apartado 4.1.2 anterior) del set a la entrada de esta etapa ($S'_t = \left\{ \bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)} \right\}_{i=1}^n$) es suficientemente alto. En caso de ser así, el paso de selección del PF no se ejecuta, lo que conlleva una importante reducción del t_{exe} , que es la razón con la que los autores de los artículos mencionados justifican el uso de paso de “*resampling adaptativo*”. El límite estandarizado entre la comunidad científica para establecer si el número aproximado de partículas eficaces es suficientemente alto es de $\hat{n}_{eff,th} = \frac{2}{3} \cdot n$.

La etapa de selección no puede ejecutarse de forma selectiva en el XPF, pues la reducción del número de partículas a la salida del estimador que se lleva a cabo en esta etapa ($n \rightarrow n - n_m$) es imprescindible en la correcta funcionalidad del filtro, como se explica en párrafos anteriores.

Finalmente, conviene comentar que el proceso de resampling propuesto en [Koller-Meier01] para el XPF es el multinomial explicando en el apartado 4.2.1 anterior. En la aplicación del filtro bayesiano a la tarea de MTT de interés se han probado varias de las alternativas de re-muestreo descritas en el mencionado apartado, llegando a la conclusión ya expuesta de que la eficiencia del filtro resultante no se modifica de manera apreciable en función del algoritmo elegido para realizar esta tarea. Las pruebas que permiten llegar a esta conclusión se incluyen en el capítulo 6, así como la definición específica del algoritmo finalmente empleado en la etapa de resampling del XPFCP usado como estimador multimodal en la tesis.

4.3.2. Puntos débiles del XPF

La robustez y la fiabilidad del XPF propuesto en [Koller-Meier01] y analizado en los puntos anteriores no se pueden asegurar, por lo que el algoritmo no puede usarse como estimador en la tarea de MTT de interés, pues ambos factores son especificaciones fundamentales en esta tesis.

Sin embargo, las características de multimodalidad, flexibilidad y sencillez (que facilitan su ejecución en tiempo real) que incluye, lo hacen especialmente interesante como estimador en la aplicación de interés, como se comenta en el punto anterior.

Debido a lo anterior, y como ya se introduce en el capítulo 3, el filtro de Bayes propuesto en esta tesis (el XPFCP) se obtiene reforzando los puntos débiles del XPF, de modo que alcance todas las especificaciones planteadas.

Por todo ello, antes de pasar a analizar en detalle la funcionalidad del XPFCP en los capítulos siguientes, se describen a continuación los puntos débiles del XPF, y la forma en que se aborda su resolución en la tesis.

a) En el paso de re-inicialización:

Como se comenta en el punto anterior, el paso de re-inicialización tiene dos funciones principales en el proceso de estimación multimodal que desarrolla el XPF: la de adaptar en el instante t la creencia procedente del anterior $t-1$ $p(\bar{x}_{t-1}|\bar{y}_{1:t-1}) \Rightarrow S_{t-1}$ a un número variable de hipótesis $k_{out,t}$ que se desea estimar (característica más importante del filtro para su empleo en la aplicación de interés), y la de eliminar el problema de empobrecimiento del set S_t , propio del algoritmo “*Bootstrap*” que se usa como base del XPF.

Para desarrollar ambas funciones, en este paso previo al de predicción se insertan n_m partículas en el set S_{t-1} , extraídas a partir del conjunto de medidas Y_t . Pero estas funciones sólo se realizan correctamente si las partículas añadidas ($\hat{s}_{i,t-1}/i = n - n_m : n$) son suficientemente representativas de las hipótesis que han aparecido en $t-1$ y que por ello no están caracterizadas en la creencia $p(\bar{x}_{t-1}|\bar{y}_{1:t-1})$ que genera el XPF en dicho instante. El inconveniente de la propuesta presentada en [Koller-Meier01] que hace que el XPF no se haya encontrado aplicado en más publicaciones, es que las n_m partículas que se insertan en la etapa de re-inicialización se obtienen a partir de un muestreo aleatorio de la densidad $p(\bar{y}_{t-1}) \Rightarrow Y_{t-1}$, por lo que no cumplen la condición expuesta (ver la Figura 4.5).

En la aplicación de MTT de interés, como la densidad $p(\bar{y}_{t-1})$ está desigualmente distribuida entre los distintos objetos a seguir, los pobremente sentidos tienen poca probabilidad de encontrarse caracterizados en el set de partículas insertadas en el paso de re-inicialización ($\hat{s}_{i,t-1}/i = n - n_m : n$); a no ser que se aumente mucho el valor del parámetro γ , disminuyendo por ende el rigor matemático del XPF, o el número total de partículas n , con el aumento del t_{exe} del algoritmo que ello conlleva.

Además, si las partículas insertadas en la etapa de re-inicialización no se eligen de forma inteligente y parcialmente determinística, el empobrecimiento de S_t seguirá produciéndose. En la aplicación de seguimiento, los objetos mejor sentidos serán representados por picos de

probabilidad en la creencia multimodal más reforzados al iterar el XPF, frente a los que representen a los pobremente sentidos, que se debilitarán hasta desaparecer de la creencia.

Este problema se muestra de forma clara en la Figura 4.6, en el modo de la creencia que aparece en los valores intermedios de x al final del proceso allí representado. Este modo está formado únicamente por una partícula procedente de las insertadas en la etapa de re-inicialización, tal y como revela el color verde del punto que la representa. Como se comenta en el apartado anterior, el peso $w_t^{(i)} / i = n - n_m : n$ de las partículas añadidas en el paso de re-inicialización $\hat{s}_{i,t-1}$ no es especialmente bajo tras el de corrección, gracias a su semejanza con la verosimilitud $p(y_t | x_t)$ que se usa para calcular $w_t^{(i)}$ (ver ecuación <4.45>).

Sin embargo, debido a la alta probabilidad de los picos que aparecen en la función de pesado $w(x_{0:t}) \rightarrow w_t^{(i)} / i = n - n_m : n$ de la Figura 4.6, representando a otros modos en la creencia, la probabilidad del pico central de esta PDF es relativamente baja, y por ello, tras el paso de selección, el número de partículas que se le asignan a la hipótesis nueva es menor que el asignado a las otras dos hipótesis (1 frente a 3 o 6), empobreciéndose así la multimodalidad de $p(x_t | y_{1:t})$ a la salida del ejemplo. De seguir así, la hipótesis nueva terminaría por desaparecer en alguna iteración posterior del XPF.

Para resolver este problema, tal y como se apunta en el capítulo anterior, en esta tesis se propone reforzar el XPF con un proceso de clasificación del set de medidas $Y_t \Rightarrow G_{1:k,t|in}$, cuya salida sea usada en la etapa de re-inicialización comentada (que se realiza entonces usando $p(G_{1:k,t-1|in})$ en lugar de $p(\bar{y}_{t-1})$), de modo que se robustezca su funcionalidad y evite el recurrente problema de empobrecimiento. La funcionalidad y los resultados de eficiencia y robustez del algoritmo así modificado, se presentan en el capítulo 6.

Además, en dicho capítulo se confirma que no es necesario fijar un valor muy alto para el parámetro $\gamma = \frac{n_m}{n}$, pues, gracias al uso de la salida del clasificador $G_{1:k,t|in}$ en distintas etapas del XPF, las n_m partículas añadidas al set S_{t-1} en el paso de re-inicialización no serán eliminadas sino muy probablemente reproducidas en el de resampling.

b) En el paso de corrección:

El proceso propuesto en [Koller-Meier01] para llevar a cabo el paso de corrección del XPF explicado en los párrafos anteriores, tiene una serie de inconvenientes que se listan a continuación para su análisis en capítulos posteriores:

- El tiempo de cómputo del conjunto de pesos $\bar{w}_t = [\tilde{w}_t^{(1:n)}]$ es proporcional al producto del número total de partículas n del XPF y el número de medidas m_t incluidas en cada instante en Y_t . El primero de los factores no puede evitarse y es común a todas las alternativas de “importance sampling” planteadas por la comunidad científica. Sin embargo, la dependencia de este tiempo de ejecución del segundo factor es especialmente problemática

en la tarea de MTT de interés, en la que, como se explica en capítulos anteriores, el conjunto de medidas de posición Y_t que se usa en el proceso de estimación se obtiene de un sistema de visión, por lo que el número m_t es generalmente elevado, y por ende lo es el t_{exe} de este paso de corrección.

- La función de verosimilitud $p(\bar{y}_t | \bar{x}_t^{(1:n)})$, usada para el cálculo de los pesos $\bar{w}_t = [\tilde{w}_t^{(1:n)}]$ (ecuación <4.45>), no es adecuada para el robusto funcionamiento del XPF. El paso de corrección así definido tiene el mismo efecto de empobrecer el set de salida S_t' que el de re-inicialización antes descrito, por las mismas razones allí expuestas: el pesado de las partículas favorece a aquellas hipótesis de seguimiento más robustas y exactamente medidas en la aplicación del XPF a la tarea de tracking de interés, frente a aquellas pobremente sensadas ([Koller-Meier01]). De este modo, las partículas relacionadas con los objetos peor sensados tienen baja probabilidad de ser mantenidas tras la etapa de selección que sigue al de corrección produciéndose así el empobrecimiento del set.

Este último efecto se observa en la Figura 4.6. La verosimilitud $p(\bar{y}_t | \bar{x}_t^{(1:n)})$ usada en el paso de corrección, produce el bajo pesado de la partícula de valor x intermedio, es decir empobrece la creencia discreta $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$ a la salida del ejemplo mostrado. De hecho este empobrecimiento puede desembocar en la desaparición de la hipótesis de estimación que esta partícula representa en $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$, como se comenta en el apartado anterior. Por otro lado, el incremento del número de partículas introducidas en la etapa de re-inicialización n_m no puede usarse como medida de mejora en este caso, pues genera nuevos problemas ya analizados.

La solución propuesta en la tesis para robustecer el paso de corrección del XPF consiste en usar el ya mencionado proceso de clasificación de medidas $Y_t \Rightarrow G_{1:k,t|in}$ en la etapa de pesado, como se adelanta en el capítulo anterior. El uso de alguna característica del conjunto de clases $G_{1:k,t|in}$ que represente de forma robusta a las medidas Y_t , filtrando así además su inexactitud y inestable distribución, mejora la fiabilidad del paso y elimina el empobrecimiento del set a su salida $S_t' = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$, como se demuestra en el capítulo 6.

Finalmente, el empleo de $G_{1:k,t|in}$ en la verosimilitud $p(\bar{y}_t | \bar{x}_t^{(1:n)})$, produce implícitamente una asociación entre cada una de las partículas del set $S_t = \{\bar{s}_{i,t} / i=1:n\}$, que caracterizan de forma estocástica a las distintas hipótesis de seguimiento incluidas en la creencia multimodal $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$ a la salida del filtro; y las clases de medidas $G_{1:k,t|in}$ que se obtienen de forma determinística con el proceso de clasificación.

Las propuestas de clasificación que se exponen en el capítulo siguiente incluyen de forma implícita además una asociación de tipo NN entre las clases de medidas y las hipótesis de targets de las que se obtienen. El uso de estas clases en la etapa de corrección completa el

proceso de asociación implícito en los clasificadores. Este proceso de asociación se analiza en mayor detalle también en el capítulo 6.

4.4. Recapitulación y conclusiones

En este capítulo se analiza el funcionamiento del PF de forma detallada, comentando sus puntos débiles y las alternativas propuestas por la comunidad científica para compensarlos.

A modo de resumen de lo aquí expuesto se puede decir que la implementación de la versión muestreada del filtro bayesiano presenta problemas de degeneración, debidos fundamentalmente a la propagación del set de muestras (partículas) $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ de una a otra iteración del algoritmo. La solución más inmediata a este problema consiste en aumentar el número de partículas del PF n . Sin embargo esta técnica conlleva un incremento directo del t_{exe} del filtro, por lo que se pierde la ventaja más importante del uso de la versión muestreada de la creencia $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$ en el proceso de estimación bayesiano.

Para resolver estos inconvenientes de ejecución del algoritmo, la comunidad científica propone otras alternativas más depuradas. Las más importantes se explican en detalle en este capítulo y pueden agruparse en dos:

1. La incorporación de un *paso de selección de partículas*, tras los de predicción y corrección típicos de cualquier algoritmo de filtrado, que minimice la varianza de $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ manteniendo la diversidad de éstas. Varios de los algoritmos propuestos en la literatura relacionada para llevar a cabo esta tarea se describen aquí (como el resampling multinomial –[Gordon93]– o el residual –[Liu98]–), si bien como se demuestra más adelante en la tesis, la eficiencia del filtro bayesiano no depende tanto del algoritmo de re-muestreo elegido como de la existencia de esta etapa en el flujograma de funcionamiento del PF.
2. La modificación del proceso de “*muestreo ponderado*” (“*importance sampling*”) del filtro bayesiano, de modo que se aproxime lo más posible al óptimo (ver ecuación <4.23>). En este punto, las soluciones propuestas por la comunidad científica pasan por el cálculo de una forma más adecuada para la función de aproximación a la creencia ($q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$ empleada en la ponderación) (como en el caso del APF –[Pitt99]–, o del “*Muestreo Segmentado*” –[MacCormick00]–), utilizando incluso estimadores complementarios para el cálculo de esta PDF (como en el caso del EKPF y el UPF –[Merwe01]–, o del “*ICondensation*” –[Isard98b]–, descritos en el apéndice I); por la modificación directa de $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ a la salida del paso de ponderación del PF (corrección) o de algún otro (como es el caso del paso MCMC –[Liu98]–, o del proceso de “*Edición Previa*” –[Gordon93]–); o por la obtención de una forma continua de la creencia $p(\bar{x}_t | \bar{y}_{1:t})$ que es propagada a la siguiente iteración del algoritmo en lugar de su forma muestreada $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$, representada por $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ (como en el GPF y el GSPF –[Kotecha03]–, o del RPF –[Doucet01]–, descritos en el apéndice I).

El objetivo de diseño de las propuestas es, en todos los casos, mantener la ventaja de implementación en tiempo real que aporta la solución muestreada del filtro de Bayes recursivo, minimizando el número de partículas del filtro n necesarias para asegurar la convergencia de la estimación en el proceso iterativo. Sin embargo, la aplicación al PF de la mayoría de las técnicas mencionadas, si bien permite alcanzar este fin, incrementa el t_{exe} del algoritmo resultante, debido a la complejidad computacional de los procesos implicados.

Por esta razón, en esta tesis se plantea el uso de una versión del PF simple y muy usada por la comunidad científica: el algoritmo “*Bootstrap*”, cuya extensión a la tarea de estimación multimodal abordada da lugar al XPF descrito en [Koller-Meier01], pero que, tal y como sus propios autores explican, presenta problemas de robustez que lo inutilizan en su aplicación real.

En la última sección de este capítulo se analiza, por tanto, la funcionalidad del XPF y sus puntos débiles, examinando la utilidad de las alternativas de la comunidad científica, antes expuestas, en la mejora de la funcionalidad del XPF.

Las conclusiones fundamentales de dicho análisis, que permiten completar el diseño del algoritmo XPFCP propuesto, se resumen a continuación:

- El XPF se elige como el algoritmo más adecuado para usar como punto de partida en el diseño de la propuesta de estimación. Esta elección se debe a que el XPF es una de las pocas soluciones aportadas por la comunidad científica en la que la multimodalidad del PF se emplea para realizar la estimación del vector de estado de un número variable de sistemas $k_{out,t}$ (ver el capítulo 2). Este algoritmo cumple, por tanto, la especificación más importante planteada en la tesis para el diseño del estimador implicado en la tarea de MTT. La multimodalidad en el XPF se consigue, fundamentalmente, con la incorporación de una nueva etapa en el flujograma del PF básico: *la de re-inicialización*.
- Por otra parte el XPF sufre los problemas de degeneración y empobrecimiento de $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$, característicos del algoritmo “*Bootstrap*” del que se deriva. Las soluciones a este problema que se proponen en la literatura relacionada y se analizan en este capítulo no son las más adecuadas para el proceso de estimación buscado, debido en parte a su alta carga computacional (por ejemplo en el caso del UPF o del paso MCMC), en parte a que no pueden ser aplicadas en este caso (como el algoritmo “*ICondensation*”), y en parte a que no aportan a la creencia multimodal obtenida a la salida del PF $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$ la robustez que requiere para ser usada en la tarea de tracking. Esta última causa se debe a la generalidad de estas soluciones, y no significa que las ideas que implican sean igualmente inválidas en el diseño del estimador multimodal. De hecho, la funcionalidad de alguna de ellas, como la del paso de “*Edición Previa*” o la del “*Muestreo Segmentado*”, coincide con la de las modificaciones que finalmente se aplican al XPF para mejorar su robustez, y obtener el XPFCP propuesto.
- Finalmente, tal y como los propios autores del XPF explican en [Koller-Meier01], la robustez del algoritmo es muy sensible a la fiabilidad de las medidas Y_t y a su distribución

homogénea entre los sistemas implicados en la estimación (correspondientes con targets a seguir en la aplicación de interés). El análisis general expuesto en el capítulo 1 indica que el conjunto Y_t , extraído de los objetos a seguir, no cumple estas especificaciones. Debido a ello, el XPF según se describe en [Koller-Meier01] no es aplicable a la tarea de estimación multimodal perseguida. Sin embargo, la combinación de este algoritmo probabilístico con uno determinístico de clasificación del set de medidas $Y_t \Rightarrow G_{1:k,t|in}$, como se introduce en el capítulo 3, permite alcanzar las especificaciones de robustez, fiabilidad, multimodalidad y tiempo real perseguidas, para dar lugar a la propuesta de filtrado de la tesis: el XPFCP.

En el paso de re-inicialización del XPF, la selección determinística del conjunto $\hat{s}_{i,t-1} / i = n - n_m : n$ a insertar en el set que caracteriza a la creencia a la entrada del filtro $\hat{S}_{t-1} = \left\{ \hat{x}_{t-1}^{(i)}, 1/n \right\}_{i=1}^n \cong \hat{p}(\bar{x}_t^{(1:n)} | \bar{y}_{1:t-1})$, elimina el problema de falta de homogeneidad de Y_t .

En el paso de corrección del XPF, el uso de las características determinísticas extraídas de la segmentación del set de medidas ($Y_t \Rightarrow G_{1:k,t|in}$) robustece la ponderación de las partículas insensibilizando la estimación de la inexactitud de este conjunto Y_t .

- Además, la combinación de los algoritmos probabilístico (el XPF) y determinístico (el clasificador de medidas $Y_t \Rightarrow G_{1:k,t|in}$) propuestos resuelve de forma implícita el problema de asociación propio, tal y como se explica en el capítulo 2, de las tareas de MTT como la de interés. La propuesta de usar un modelo sencillo (de tipo CV) en la estimación de movimiento de los sistemas implicados en la tarea de seguimiento, facilita el desarrollo del proceso de asociación implícita.

A partir de estas conclusiones, en los capítulos siguientes se describe: primero (en el capítulo 5) la propuesta de segmentación determinística planteada en la tesis para ser combinada con el algoritmo XPF analizado y dar lugar al XPFCP; y después (en el capítulo 6) la funcionalidad completa y detallada del XPFCP, que constituye así la propuesta de estimación multimodal planteada en la tesis para resolver el problema de tracking expuesto en el capítulo 1.

CAPÍTULO 5

CONTENIDO DETERMINÍSTICO DE LA PROPUESTA

Tras analizar en el capítulo anterior el marco probabilístico de la solución de MTT propuesta, en este capítulo se describe el marco determinístico de la misma, que completará el algoritmo de seguimiento combinado (XPFCP) expuesto de forma global en el capítulo 3.

De la revisión de trabajos realizada en el capítulo 2, se concluye que la parte que todavía se encuentra sin solución evidente en la tarea de tracking de múltiples objetos es la de asociar cada medida obtenida del entorno con el target concreto del que ha sido extraída. Las propuestas de asociación utilizadas por distintos autores en los trabajos revisados pueden agruparse en dos grandes bloques: técnicas estocásticas (como la del JPDAF) y determinísticas (como las técnicas de segmentación), menos usadas que las anteriores.

En esta tesis se aborda la resolución del problema de asociación mediante métodos determinísticos, pues como se expone en el capítulo 4, aportan al estimador probabilístico utilizado en la propuesta (el PF) una exactitud que la variable estocástica no tiene. Además, la alternativa determinística que se propone en esta tesis proporciona al estimador multimodal la robustez necesaria para incorporar nuevos objetos en el proceso de seguimiento con suficiente fiabilidad como para que las nuevas hipótesis de estimación permanezcan en la ejecución iterativa del estimador, a pesar de su baja verosimilitud inicial. Por estas razones se propone el uso de un clasificador de medidas para desarrollar las tareas comentadas.

En el capítulo 3, por otra parte, se expone la necesidad de obtener una salida determinística de la PDF multimodal y discreta que genera el XPFCP, que indique, en su aplicación a la tarea de tracking de interés, el número de objetos seguidos en cada instante. Para ello allí se propone el uso de otro proceso de clasificación de partículas, de diferentes especificaciones al anterior.

En este capítulo se describen dos segmentadores diseñados con estos objetivos: uno de ellos basado en el estándar “*k-medias*” (también conocido en lengua inglesa como “*k-means*”) y otro basado en el algoritmo borroso “*subtractive*”. La versión que de cada uno de ellos se ha diseñado en la tesis, se denomina respectivamente “*k-medias secuencial*”, y “*subtractive enlazado*”, por las razones que se exponen más adelante en el capítulo.

La razón que lleva a elegir los algoritmos “*k-medias*” y “*subtractive*” para su valoración como métodos de asociación en la tarea de seguimiento de múltiples targets propuesto, es la de hacer lo más completo posible el análisis de las aportaciones que pueden incorporar a éste dos clasificadores muy diferentes. En el apéndice II se describen las características de uso más frecuente a la hora de definir un segmentador, y que serán referidas a lo largo de este capítulo. Además, en el mismo anexo se presentan algunos de los parámetros más usados por la comunidad científica para comparar de forma sistemática y cuantitativa las distintas alternativas de clasificación, y que se emplean en la tesis para validar los clasificadores propuestos.

En el diseño de estos algoritmos se ha teniendo en cuenta su función final en el sistema de MTT, incorporándose en su diseño una serie de funcionalidades y parámetros que serán usados en las distintas etapas del estimador multimodal propuesto en la tesis. Para facilitar la descripción de los clasificadores propuestos, en este capítulo se analiza su funcionamiento como segmentadores de medidas, dejando para el capítulo siguiente el análisis de las consideraciones a tener en cuenta en su aplicación a la clasificación de partículas.

En esta parte de la memoria se analiza la funcionalidad de cada uno de los dos clasificadores, y se ejemplifica la misma en distintos experimentos y comparativas, que permitan concluir la idoneidad de cada uno de ellos para las tareas comentadas.

5.1. Algoritmos de clasificación

El concepto general de clasificación se refiere a la búsqueda de una organización dentro de un conjunto de datos en función de una o más “*características*”, en un “*espacio de definición*” fijado por éstas. Realizada la organización, a cada una de las agrupaciones resultantes se le denomina “*clase*” o “*grupo*”, mientras que al set de datos que son asociados a cada clase se les suele llamar “*miembros de una clase*”. En este punto se especifican estos conceptos básicos de funcionamiento de los clasificadores en su aplicación a la tarea de interés.

Los segmentadores se encuadran dentro del área de conocimiento denominada “*Minería de Datos*” (“*Data Mining*”) debido al objetivo común de estas técnicas de extraer la información relevante oculta en la organización de los datos. Es una ciencia en gran expansión ya que se encuentra directamente relacionada con la inteligencia artificial y la tecnología de la información. Dentro de esta área de conocimiento se distinguen fundamentalmente dos líneas de

trabajo: la de aprendizaje sin supervisión (“*Indirect Mining*”), que incluye a los procesos dirigidos al modelado de sistemas en base al análisis de los datos que arrojan como salida ante ciertas entradas; y la de clasificación (“*Direct Mining*”), que incluye las técnicas relacionadas con la organización de datos, la compresión de la información (bases de datos), etc.

Evidentemente, teniendo en cuenta la variedad de las aplicaciones de los segmentadores, existen multitud de alternativas propuestas por distintos autores para distintos tipos de aplicaciones. En cualquier caso y debido a la misma razón, normalmente es necesario diseñar el clasificador que más se adecua a cada aplicación concreta.

En el apéndice II se incluye un análisis de los tipos más importantes de procesos de clasificación descritos en la literatura científica, así como las características que los individualizan. El estudio allí expuesto se ha llevado a cabo con el objetivo de poder decidir con criterio el algoritmo más adecuado para la aplicación de MTT.

Parece relevante comentar que en este documento se van a utilizar como sinónimos del término clasificación el de agrupación y el de segmentación, si bien en la literatura del área relacionada, sobre todo en la inglesa, se diferencia entre “*segmentation*” (segmentación), “*clustering*” (agrupación), y “*classification*” (clasificación). Los dos primeros términos están más relacionado con la tarea que aquí se propone, la de encontrar una organización dentro del conjunto de datos sin ningún conocimiento a priori sobre el número o las características específicas de cada grupo; mientras que el término de clasificación se asocia generalmente con la ordenación de información previamente etiquetada (a partir de un conocimiento previo sobre la identidad de los datos a organizar y las clases).

5.1.1. Nomenclatura de clasificación

Para facilitar la lectura de este capítulo, en este apartado se resume la nomenclatura relacionada con la clasificación. La organización de clases resultado de la segmentación da lugar a un set k de grupos $G_{1:k}$, definidos, en principio, como se muestra:

$$G_{j=1:k} = \{\bar{g}_j, \bar{l}_{1:l_j} / j=1:k\} = \{\bar{g}_j, L_j / j=1:k\}^1, \quad <5.1>$$

donde $L_j = \{\bar{l}_{i,j} / i=1:l_j\}$ referencia al conjunto de l_j miembros que el clasificador asocia a la clase j . Además el set de datos a clasificar, del que L_j es un subconjunto, se va a referenciar a través de la matriz $Y = \{\bar{y}_i / i=1:m\}$. Por otro lado \bar{g}_j es el vector característico que identifica a la clase j . Este vector generalmente se define en el mismo espacio de definición que Y , pudiendo de hecho, ser o no uno de los datos del set. Normalmente, \bar{g}_j suele ser el centroide de L_j , en el espacio de definición elegido, por lo que se le denomina “*centroide*”.

¹ En este capítulo se habla de forma genérica de clases para referenciar a clases de medidas, sin incluir el subíndice *in* que permite realizar esta especificación (ver capítulo 3). En el capítulo 6 se establecen las consideraciones a tener en cuenta en la aplicación de estos segmentadores a la clasificación de partículas, especificada con el subíndice *out*.

La aplicación de las variables anteriores al proceso de segmentación de medidas de interés en esta tesis, permite definir cada dato a ordenar como una medida de posición en un espacio tridimensional cartesiano $\vec{y}_i = [x \ y \ z]$, tal y como se presenta ya en el capítulo 2 de esta memoria. Todos los datos a clasificar son, por tanto, puntos en ese entorno tridimensional y cartesiano (XYZ), y los grupos creados ($G_{1:k}$), así como los parámetros que las describen ($G_{1:k} = \{\vec{g}_j, L_j\}_{j=1}^k$) se definen también en el espacio XYZ , por lo que los miembros de cada grupo ($L_j = \{\vec{y}_{i,j} / i=1:l_j\}$), y su centroide (\vec{g}_j), son también vectores tridimensionales.

La clasificación de partículas también incluida en el XPFCP propuesto en la tesis se desarrolla, por el contrario, en el espacio de cinco dimensiones en el que éstas se definen. En el capítulo siguiente se analizan las consideraciones a tener en cuenta en ese caso, restringiéndose el estudio realizado en este capítulo a la clasificación en el espacio XYZ .

5.1.2. Espacio de clasificación

Al margen del algoritmo, aunque directamente relacionada con éste, la especificación más importante para realizar la clasificación es la o las características comparativas y discriminantes que se eligen para distinguir los diferentes grupos dentro de Y .

Esta variable discriminante se referencia en la literatura científica simplemente como “*característica*” o “*vector de características*”, y de ella depende la robustez y la fiabilidad del clasificador, incluso en mayor medida que del tipo de algoritmo usado.

Por otro lado, el cálculo de esta variable puede complicar la ejecución del segmentador, ya que en muchos casos implica realizar una transformación del espacio en el que se define Y al de definición del vector de características. El espacio de características más adecuado dependerá del tipo de datos a ordenar (métrico para datos de posicionamiento, de color para imágenes, etc.). Este proceso llega a consumir recursos, en general, y tiempo, en particular, inaceptables para aplicaciones de tiempo real, como la que se resuelve en esta tesis.

Realizada la transformación del espacio, la segmentación se lleva a cabo empleando una medida de “*distancia*” normalmente entre \vec{g}_j de cada clase $G_j / j=1:k$ y los datos candidatos a pertenecer a la clase L_j , o también entre cada par de datos de Y , \vec{y}_1 e \vec{y}_2 .

Cuando se utiliza la distancia como elemento de clustering se habla de “*métodos de segmentación basados en distancia*” (“*distance based clustering*”), y de ese modo se diferencian de los “*métodos de segmentación conceptual*” (“*conceptual clustering*”) ([Berkhin02]), basados en un concepto descriptivo más abstracto para realizar la organización. La aplicación de interés está más relacionada con la primera alternativa que con la segunda.

El espacio de características elegido para calcular las medidas de distancia tendrá tantas dimensiones como tamaño tenga el vector de características que se elija para hacer el clustering. Teniendo en cuenta que, en la aplicación del clasificador al conjunto de medidas Y , el espacio de definición de éste es el cartesiano tridimensional XYZ , parece evidente elegir el mismo

espacio para la definición de la característica de distancia. El uso de otro espacio complicaría la segmentación, pues requeriría una etapa de transformación previa del conjunto de datos, y, siguiendo la regla básica de diseño de clasificadores, posiblemente no mejoraría el resultado.

Por otro lado, analizando en detalle el contenido del vector de posición \vec{y}_i que define cada dato i del set a organizar Y , y teniendo en cuenta la aplicación de interés, es fácil concluir que la dimensión de altura (y) no aporta información útil en la clasificación. La Figura 5.1 muestra la representación tridimensional en el espacio XYZ de la matriz Y , obtenida en un experimento real con el sistema de estereo-visión descrito en el capítulo 1 y en [Marrón06a] o [Marrón07a], así como su proyección bidimensional en el plano XZ . En la Figura 5.2 se muestra la imagen real de la que se han tomado los datos de la matriz Y .

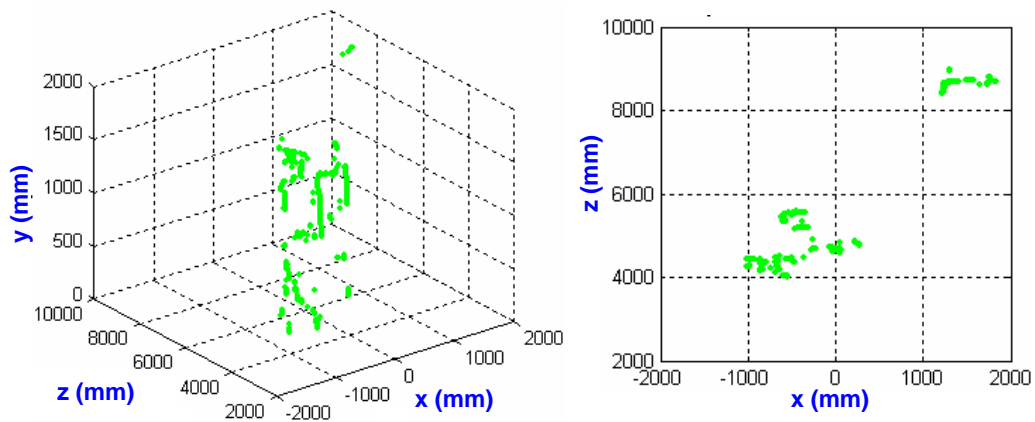


Figura 5.1. Representación 3D en el espacio XZY (a la izquierda) y proyección 2D en el plano XZ (a la derecha) de un set de datos Y extraído de un experimento real en el que hay cuatro objetos en una escena.



Figura 5.2. Imagen de la escena real de la que se extrae el set de medidas Y representado en la Figura 5.1. La proyección del conjunto en el plano imagen se muestra en verde.

A la vista de la Figura 5.1 y la Figura 5.2, es evidente que la información que permite decidir con qué objeto de la escena se relaciona cada punto de Y se encuentra en las coordenadas de su proyección XZ . De hecho la coordenada de altura (y) en este espacio puede dificultar la clasificación pues esta variable no define inequívocamente la asociación de cada \vec{y}_i con un target de la escena. Por todo ello, la variable de distancia que se utiliza en la segmentación de medidas usado en algoritmo XPFCEP se define en el espacio cartesiano bidimensional XZ .

Una vez definido el espacio de medida de la distancia, es necesario elegir la forma que se desea utilizar para esta variable. En el apéndice II se analizan las más típicas, y se justifica el uso de la “*distancia euclídea*” $d_{euclídea|i,j}$ como base del proceso de clasificación de interés.

5.2. El algoritmo “*k-medias secuencial*”

El “*k-medias secuencial*”, basa su funcionamiento en el estándar “*k-medias*” (conocido en inglés como “*k-means*”) que es un algoritmo de clasificación recursivo, enlazado y booleano.

La definición práctica del estándar “*k-medias*” fue inicialmente presentada en [MacQueen67], donde se plantean un conjunto de reglas que establecen su funcionamiento:

- Los elementos pertenecientes a cada grupo deben ser similares entre sí, y diferentes a los de otros grupos, en lo referente a la característica diferencial elegida para realizar el clustering.
- Los datos se asocian a un sólo grupo (organización unívoca, booleana y enlazada).
- No puede quedar ninguna clase vacía.

La ventaja más importante del estándar “*k-medias*” es su bajo t_{exe} en una iteración, debido a su baja complejidad computacional, si bien su carácter recursivo puede producir la no convergencia del proceso si no existe una solución única de segmentación, en cuyo caso su t_{exe} se dispara. Para resolver este problema es necesario establecer siempre un límite de iteraciones de modo que se detenga la ejecución del algoritmo.

El estándar “*k-medias*” es también, en primera instancia, un proceso no repetible. Su recursividad resuelve, sin embargo, este problema, a costa del incremento en su t_{exe} .

Por otro lado, la versión básica de este segmentador está pensada para organizar el set Y en un número fijo k de clases. Será necesario modificar el proceso básico para poder adaptarlo a su ejecución para un número inicialmente desconocido de clases, obteniendo como resultado el algoritmo conocido como “*k-medias extendido*” (“*extended k-means*” en inglés).

Finalmente en la tesis se plantean una serie de mejoras al “*k-medias extendido*” que incrementan la robustez y fiabilidad del clasificador, dando lugar al “*k-medias secuencial*”.

5.2.1. El “*k-medias*” básico

A partir de las reglas de definición del estándar “*k-medias*” expuestas, la versión básica del mismo se describe a través de las siguientes etapas ([Alsabti98] o [Kanungo02]):

1. Elegir k datos del set Y para utilizarlos como valor inicial de los centroides ($\vec{g}_{0,j} / j = 1 : k$) de los clusters a crear.

2. Calcular el valor de distancia de cada dato ($\vec{y}_i / i = 1:m$) a todos los centroides ($\vec{g}_{iter,j} / j = 1:k$): $d_{iter,i,j}$; y asignar el \vec{y}_i al cluster con el $\vec{g}_{iter,j}$ a una distancia menor.
3. Realizada la asignación de todo el set Y , se recalcula el valor del centroide de cada clase ($\vec{g}_{iter,j}$), como la media aritmética del conjunto de datos asignados a la clase ($L_{iter,j}$) en el espacio de definición de la distancia.
4. Se analiza si ha habido algún cambio en la última iteración en el centroide de cada cluster (\vec{g}_j) o, lo que es equivalente, de los miembros asignados a cada cluster (L_j). Si es así, el algoritmo se repite desde el paso 2.
5. En caso contrario, o si el número de veces que se ha ejecutado el proceso excede un límite predefinido, el algoritmo finaliza, antes de lo cual se eliminan los conjuntos que han quedado sin miembros asociados ($L_j = \emptyset / j = 1:k$).

En algunas de las implementaciones del sistema expuesto en la literatura científica ([Alsabti98]) se impone como condición para su finalización que el movimiento de todos los centroides de los conjuntos $\vec{g}_{1:k}$ en el espacio de distancia sea menor a un umbral predeterminado. Este proceso es, funcionalmente hablando, equivalente al de terminación comentado anteriormente, pero tiene un coste computacional mucho mayor, pues hay que calcular en cada iteración el mencionado movimiento de los $\vec{g}_{1:k}$, por lo que su uso no se considera interesante en esta tesis.

La funcionalidad descrita en los párrafos anteriores se resume en el flujograma mostrado en la Figura 5.3. Por otro lado, en la Figura 5.4 se muestran dos ejemplos de ejecución del proceso básico para un mismo conjunto Y en un espacio bidimensional ($\lambda_T = 2$), empleando como parámetro diferenciador de grupos la distancia euclídea ($d_{euclidea(i,j)}$), y para un número de grupos $k = 2$ y $k = 3$. En la Figura 5.4, cada \vec{y}_i se presenta con un color diferente en función de el grupo con la que el segmentador lo asocia ($\vec{y}_i \in L_1$ en rojo, $\vec{y}_i \in L_2$ en verde, $\vec{y}_i \in L_3$ en magenta) y el centroide de cada grupo ($\vec{g}_{1:3}$) se muestra con un asterisco negro. Como se aprecia en las imágenes, la organización final del Y depende del número de grupos k que se especifique como entrada del proceso, y será booleana y con conexión (enlazada).

5.2.2. El “*k-medias extendido*”

El clasificador “*k-medias*” es de la familia de algoritmos del tipo BSAS (“*Basic Sequential Algorithm Scheme*” o “*Algoritmo de Esquema Secuencial Básico*”), el MBSAS (“*Modified BSAS*” o “*BSAS Modificado*”) o el TTSAS (“*Two Thresholds Sequential Algorithm Scheme*” o “*Algoritmo de Esquema Secuencial con Dos Límites*”) ([Kainulainen02]). De hecho, cada una de estas técnicas define una funcionalidad más específica que la referida con el nombre de “*k-medias*”, bajo el que se agrupan distintos comportamientos dependiendo del autor consultado.

Más concretamente, en la definición de los tres algoritmos comentados aparece una importante característica que no siempre se encuentra en las distintas referencias que se hacen

del estándar “*k-medias*”, que consiste en que el número de grupos k en las que se organiza Y en la clasificación no tiene porqué estar prefijado. Esta característica ha de ser necesariamente incluida en la definición del segmentador a usar en la aplicación de interés, pues según los objetivos planteados al principio de esta memoria, el número de objetos k de los que proceden las medidas de posición a ordenar es desconocido.

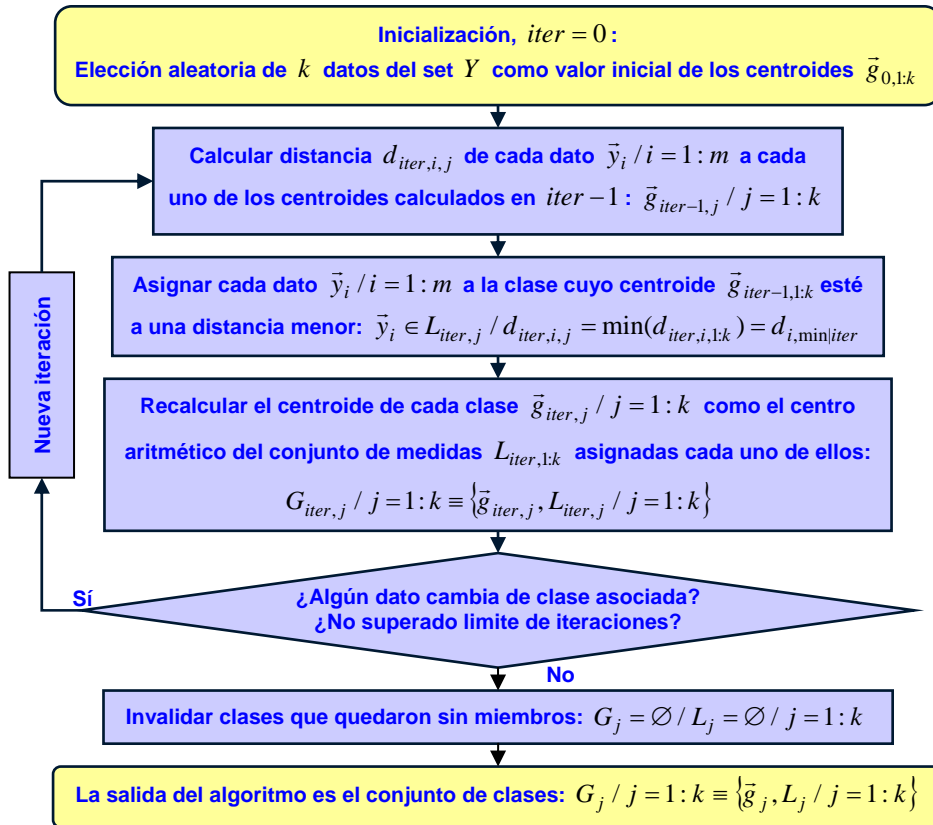


Figura 5.3. Diagrama funcional de la versión básica del “*k-medias*”.

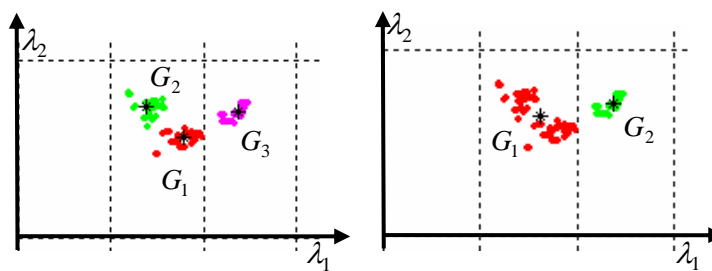


Figura 5.4. Ejemplos de ejecución del “*k-medias*” básico para un mismo set Y , en un espacio bidimensional ($\lambda_T = 2$), utilizando como parámetro diferenciador la distancia euclídea ($d_{euclidea[i,j]}$), y para un número de clases $k = 3$ (imagen izquierda) y $k = 2$ (imagen derecha).

En la Figura 5.5 se muestra la modificación de la funcionalidad del “*k-medias*” básico de modo que se adapte a esta especificación. El proceso descrito en la figura es una interpretación de las distintas versiones que se refieren, en general, con el nombre de “*k-medias extendido*”.

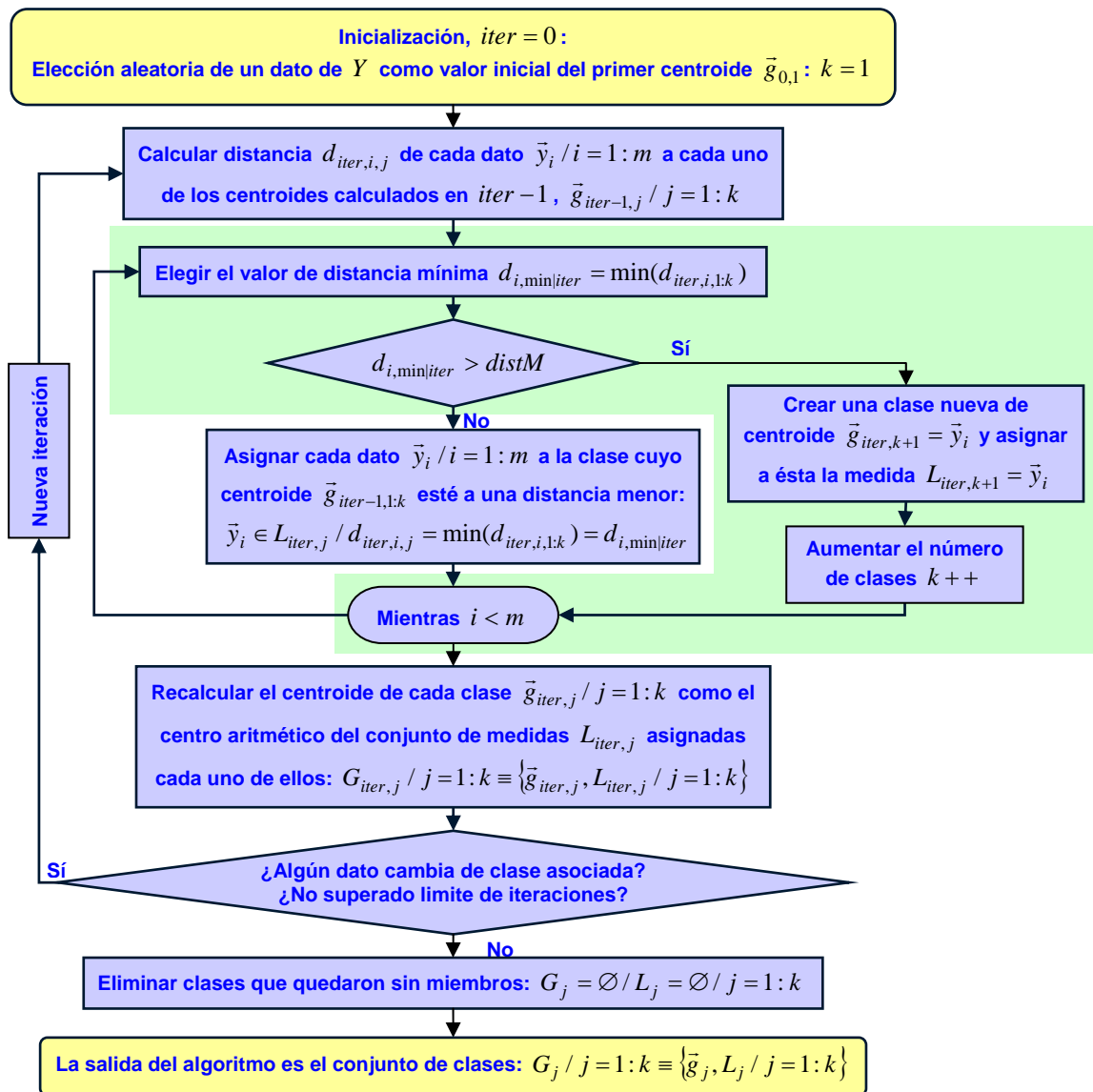


Figura 5.5. Diagrama funcional del “k-medias extendido”. Sus diferencias funcionales respecto al flujograma del “k-medias” básico se destacan sobre fondo verde.

Como se aprecia en la Figura 5.5, la parte añadida (bloques sobre fondo verde) al flujograma de la anterior Figura 5.3 permite que el algoritmo cree un nuevo grupo en el proceso de segmentación, en caso de que alguno o algunos de los datos de Y tenga una característica de distancia muy diferente a la de los grupos existentes.

La aplicación práctica de la idea expuesta en el párrafo anterior, exige la inclusión de un parámetro de distancia límite ($distM$ en la Figura 5.5) con el que se compara el valor de distancia $d_{i,min} = \min(d_{i,1:k})$ calculado para cada $\bar{y}_i / i = 1 : m$. Además la definición del parámetro $distM$ conlleva la especificación de una forma para cada conjunto en el espacio de definición de la distancia, en función del tipo de distancia (ver en el apéndice II).

La Figura 5.6 muestra el resultado de la ejecución del proceso para el mismo conjunto Y , bajo las mismas especificaciones que las mostradas en la anterior Figura 5.4. La diferencia

fundamental, frente a las condiciones establecidas para la obtención de la Figura 5.4, se encuentra en que, en este caso, no ha sido necesario fijar k , pero sí el valor del parámetro $distM$. Sobre la organización resultante se ha dibujado también, para cada clase $G_{l;3}$, el límite de distancia que fija el parámetro $distM$. Tal y como se aprecia en la Figura 5.6, este límite determina una área circular alrededor de $\bar{g}_{l;3}$ (señalados con cruces negras en la figura) dentro de la que se localizan, en el espacio de distancia empleado en la clasificación, todos los datos que se asocian a cada grupo $L_{l;3}$. La forma circular se debe al uso de la distancia euclídea ($d_{euclidean|i,j}$) como parámetro diferencial en un espacio bidimensional $\lambda_T = 2$.

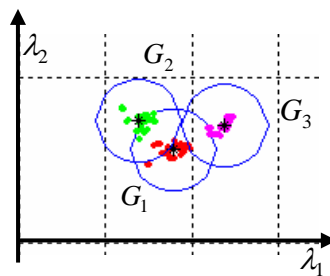


Figura 5.6. Ejemplo de ejecución del “*k-medias extendido*” en el experimento descrito en la Figura 5.4. Los círculos azules representan el valor de $distM$ en el espacio de clasificación.

El “*k-medias extendido*”, descrito en la Figura 5.5, presenta como inconveniente principal una evidente sensibilidad al valor del parámetro $distM$. En el apartado siguiente se comentan exhaustivamente, éste y otros puntos débiles de este sistema, que han llevado al diseño de una versión mejorada del “*k-medias extendido*” que se adapte a las especificaciones de la tesis.

Analizando las versiones del “*k-medias extendido*” que se proponen en las distintas fuentes literarias consultadas, se observa que solamente el algoritmo TTSAS muestra una funcionalidad verdaderamente diferente a la presentada en la Figura 5.5.

En el algoritmo TTSAS se introduce un margen de histéresis en el proceso de asignación de cada dato de Y a uno u otro conjunto, estableciéndose así dos límites de comparación para la variable distancia $distM1 < distM2$, que se usan para realizar la asignación en dos pasos:

- Uno primero en el que se resuelven las asociaciones más evidentes (las de aquellos \bar{y}_i de los que se obtiene un valor de distancia $d_{i,\min} < distM1$) y se crean las clases relacionadas con los \bar{y}_i más dispares (con un valor de distancia $d_{i,\min} > distM2$). A partir de esta organización se realiza un cálculo del valor de los centroides $\bar{g}'_{iter,1:k}$ aproximado al final.
- Uno segundo en el que cada uno de los \bar{y}_i , que en el espacio de distancia utilizado ha quedado ubicado en el margen de histéresis ($distM1 < d_{i,\min} < distM2$), se asocian a la clase cuyo centroide $\bar{g}'_{iter,1:k}$ se encuentre a una medida de distancia menor. Se realiza un segundo cálculo del valor de $\bar{g}'_{iter,1:k}$ y se continúa con la iteración siguiente.

La Figura 5.7 muestra una representación de el efecto que tienen los dos parámetros $distM1 < distM2$ en un espacio de distancia bidimensional $\lambda_T = 2$, y basado en la distancia euclídea ($d_{euclidea_{i,j}}$), como el empleado en los ejemplos de la Figura 5.4 y la Figura 5.6.

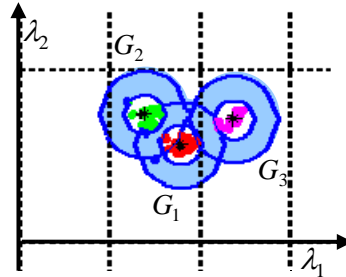


Figura 5.7. Ejemplo de ejecución del algoritmo TTSAS en el experimento descrito en la Figura 5.4. El margen de histéresis que se crea en el espacio de características para $distM1 < distM2$ se destaca en azul claro.

El objetivo de este proceso es el de disminuir el número de iteraciones del algoritmo, al calcular en una primera aproximación un valor robusto del centroide de cada grupo, pues depende solo de los \bar{y}_i que se clasifican de forma más evidente. La zona de histéresis (en tono azul claro en la Figura 5.7) en el espacio de distancias no se utiliza, sin embargo, para definir un área de pertenencia intermedia o borrosa para Y , útil como zona de intersección de grupos o de eliminación de los outliers incluidos en el conjunto.

El proceso de histéresis expuesto, podría haberse utilizado en esta tesis para modificar el “*k-medias extendido*”, descrito en la Figura 5.5, y generar una versión borrosa de éste², pues la solución más eficaz a los problemas de sensibilidad que presenta el segmentador consiste en la sustitución del mismo por un clasificador borroso como el “*subtractive*” ([Berkhin02]).

Sin embargo, se ha optado por mantener el carácter booleano de la versión del “*k-medias*” a emplear en la tarea de clustering de interés, y comparar los resultados conseguidos de este modo con los que arroja un algoritmo puramente borroso como el “*subtractive*”, expuesto en un apartado posterior. En contrapartida, los problemas de sensibilidad del algoritmo booleano son resueltos en la tesis mediante la incorporación de otros sistemas que aportan robustez al clasificador, tal y como se expone a continuación.

5.2.3. Análisis de los puntos débiles del “*k-medias extendido*”

En este apartado analizan las características del “*k-medias extendido*” que es necesario mejorar para usarlo como segmentador en la aplicación de MTT. Este análisis da lugar al desarrollo de dos aportaciones al “*k-medias*” básico:

- El algoritmo “*k-medias secuencial*”.
- El proceso de validación de las clases, aplicable también al clasificador basado en el “*subtractive*”, descrito en un apartado posterior de este capítulo.

² La versión borrosa del algoritmo “*k-medias*” es, de hecho un algoritmo ampliamente documentado en la literatura del área como “*k-medias borroso*”, o en inglés “*fuzzy c-means*”: FCM ([Ruspini69], [Dunn74] o [Bezdek81]).

Ambas aportaciones son analizadas en profundidad más adelante en este capítulo.

a) Sensibilidad frente al ruido

El carácter booleano del estándar “*k-medias*” no permite fácilmente hacer un tratamiento especial para outliers que se encuentren incluidos en el set Y , de modo que se filtren y se eliminen de la clasificación arrojada por el segmentador, o al menos se asocien a un grupo especial ([Lesaux02]).

La solución que se aporta en esta tesis a este problema consiste en la inclusión de un proceso de validación en el segmentador, que permita, entre otras cosas, filtrar los outliers, aprovechando para ello el carácter esporádico de su aparición en el set a organizar. De este modo se aumenta la robustez del sistema, y se cumple una de las especificaciones planteadas.

En cualquier caso, en el apartado siguiente se hace un análisis comparativo del comportamiento del sistema propuesto sin validación frente al “*k-medias extendido*” en lo que a robustez frente a ruidos se refiere.

b) Sensibilidad frente a variaciones de $distM$

La falta de robustez del segmentador debida al parámetro $distM$ es evidente, y se analiza en los párrafos siguientes. De nuevo en este caso, el proceso de validación ya mencionado se usa para resolver este problema, como se demuestra en un apartado posterior.

Parece interesante incluir aquí un análisis de la sensibilidad del algoritmo respecto al parámetro $distM$, con el fin de poder comprobar, más adelante en esta memoria, que el proceso de validación propuesto realmente disminuye la sensibilidad del clasificador frente a $distM$. En función del valor de este parámetro la salida del segmentador cambia en lo referente a:

- Número de grupos k ,
- la parte de Y asociado a cada grupo L_j , y por ende, el centroide del mismo \bar{g}_j , y
- tamaño del espacio asociado al grupo.

La Figura 5.8 muestra el resultado del clustering obtenido tras ejecutar el “*k-medias extendido*” para tres valores diferentes de $distM$ ($distM1 = distM0 / 1.5 = 0.66 \cdot distM2$, $distM0$ y $distM2 = 1.5 \cdot distM0$) en el experimento descrito en la Figura 5.4. Tal y como se aprecia en ella, la modificación del valor de $distM$ afecta a los tres factores expuestos.

Los efectos comentados tienen una relación directa con las especificaciones de fiabilidad y flexibilidad planteadas en la tesis. La fiabilidad del clasificador depende fuertemente del valor de este parámetro de distancia, pues los dos primeros factores enumerados determinan las características de separación y distorsión de las clases resultantes (ver en el apéndice II). El análisis de los valores de distorsión y separación obtenidos con el “*k-medias extendido*” se incluye más adelante, para compararlos con los arrojados por el “*k-medias secuencial*”.

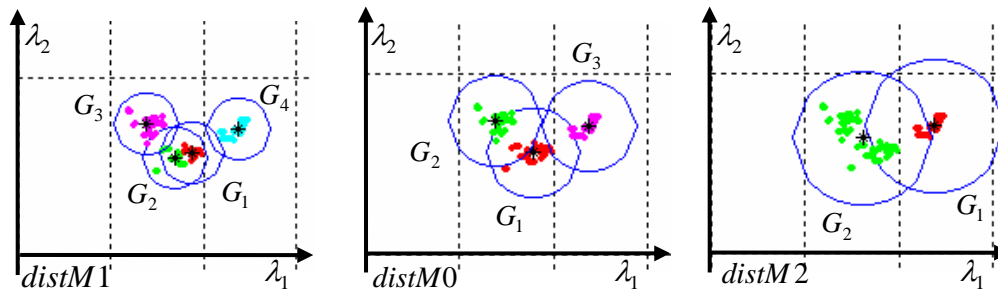


Figura 5.8. Ejemplos de ejecución del “*k-medias extendido*” en el experimento descrito en la Figura 5.4. Se muestran los resultados del algoritmo para distintos valores del parámetro $distM$: de izquierda a derecha

$$distM1 = distM0 / 1.5, \quad distM0 \quad \text{y} \quad distM2 = 1.5 \cdot distM0.$$

Por otro lado, la especificación de flexibilidad establece que los objetos implicados en la tarea de seguimiento puedan tener una tipología variada. Esto significa que los datos de posición extraídos de éstos no tienen porqué mostrar una disposición tipo, ni en forma (los objetos pueden ser incluso flexibles) ni en tamaño, aún buscando un espacio de distancias adecuado a la hora de clasificarlos.

La sensibilidad del segmentador frente a variaciones de $distM$ tiene un efecto fundamental en el incumplimiento de la especificación anterior, tal y como se explica a continuación:

- La definición de un valor constante del parámetro $distM$ restringe el clustering a situaciones en las que el conjunto L_j se circunscriba a la zona delimitada por $distM$ en el espacio de distancias, lo cual no se ajusta a la especificación mencionada. Si el valor de $distM$ es demasiado pequeño, el algoritmo organizará los datos asociados a un solo grupo (L_1) en dos ($L_2 \cup L_3 = L_1$), produciendo un problema de duplicación como el de la Figura 5.8.
- Por otro lado, el uso de un valor de $distM$ suficientemente alto como para permitir la correcta segmentación de los datos extraídos de distintos tipos de targets, no es la solución al problema. El aumento del valor de $distM$ conlleva una disminución en la distancia mínima que puede aparecer entre dos sets de datos extraídos de sendos targets, sin el riesgo de que éstos (L_1 y L_2) se asocien a un mismo grupo ($L_3 = L_1 \cup L_2$). En la Figura 5.8 se aprecia un ejemplo de este efecto negativo.

El uso de un valor adaptativo para este parámetro podría resolver los problemas de sensibilidad comentados. Sin embargo, el proceso de validación propuesto en apartados anteriores para mejorar la robustez del segmentador, puede usarse con esta misma función, evitando así complicar la funcionalidad del clasificador. En un apartado posterior se analiza la sensibilidad frente a $distM$ del “*k-medias secuencial*” con validación, justificándose entonces lo innecesario de usar un valor adaptativo para el parámetro $distM$.

La Figura 5.9 muestra la tasa de error del “*k-medias extendido*” en el experimento complejo (ver en el apéndice III) para distintos valores de $distM$. Los errores se han agrupado en los dos tipos comentados en los párrafos anteriores: errores de clases duplicadas debidos a la elección

de un valor excesivamente bajo de $distM$, y errores de unión de 2 clases en 1, debidos a un valor excesivamente alto de $distM$. De la Figura 5.9 pueden extraerse dos conclusiones claras:

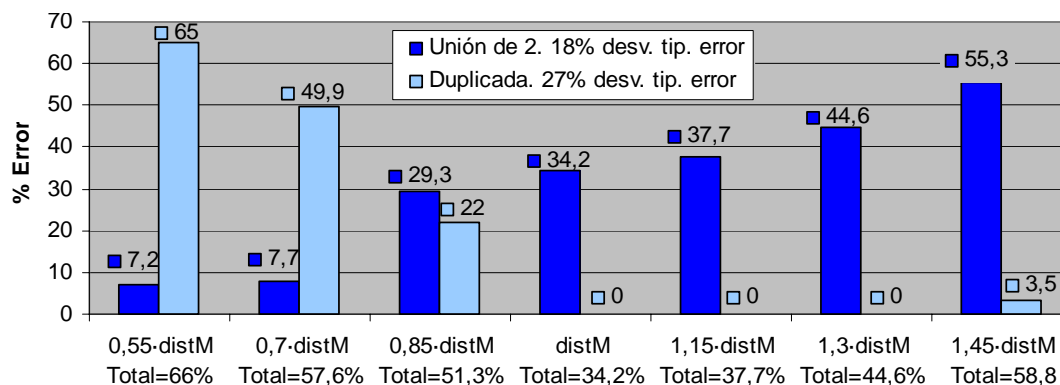


Figura 5.9. Tasa de error del “*k-medias extendido*” en el experimento complejo en función del valor de $distM$.

- La sensibilidad del “*k-medias extendido*” frente a $distM$ es evidente. Para una modificación de $\pm 50\%$ sobre el valor de ajuste del parámetro, la variación de la tasa de error sobre la media es muy grande (22% en errores de clases duplicadas y el 24% en los unión de clases).
- Tal y como se describe en el análisis funcional del parámetro, al modificar el valor de $distM$, aumenta uno de los dos tipos de errores comentados y disminuye el otro. No es posible obtener un valor de ajuste del parámetro, que permita utilizar el “*k-medias extendido*” con sets de datos de distinta tipología, como fija la especificación de flexibilidad.

Todo lo anterior confirma empíricamente la falta de robustez del “*k-medias extendido*” ante modificaciones del parámetro $distM$, y la consiguiente necesidad de introducir un proceso de validación que mejore esta característica.

c) Convergencia y tiempo de ejecución

Debido al carácter iterativo del “*k-medias extendido*”, la convergencia del mismo no se puede asegurar y, de todas formas, su t_{exe} no es fijo, sino que depende del valor asignado aleatoriamente a los centroides en la primera iteración del proceso $\bar{g}_{0,1:k}$ ([Alsabti98]). Desde un punto de vista teórico, la complejidad computacional de este algoritmo es $O(m^2)$ ([Weiss97]), teniendo en cuenta que, en caso de convergencia, el número máximo de veces que itera es igual al número de datos a clasificar (m).

Para asegurar la convergencia del proceso y minimizar su t_{exe} , diferentes autores han propuesto varias modificaciones del “*k-medias*” básico que tienen como característica común la elección no aleatoria sino determinística y orientada del valor inicial de los centroides de los grupos $\bar{g}_{0,1:k}$ (p. ej. el algoritmo “*X-means*” descrito en [Pelleg00]). Con el mismo objetivo de minimizar el t_{exe} del segmentador, y alcanzar una de las especificaciones de la tesis, se propone usar como valor inicial de los centroides su posición estimada en el espacio de distancias.

La inclusión de la mencionada etapa de predicción en el “*k-medias extendido*” da lugar al “*k-medias secuencial*”, propuesto.

5.2.4. Versión del algoritmo desarrollada en la tesis: el “*k-medias secuencial*”

A la vista del análisis funcional de los procesos expuestos en apartados anteriores, se propone utilizar como segmentador en la tarea de MTT una modificación del “*k-medias extendido*”, en la que se añade una etapa previa de estimación del valor inicial de los centroides ($\bar{g}_{0,1:k,t} = \bar{g}_{1:k,t|t-1}$), para mejorar la convergencia y disminuir el t_{exe} del segmentador. El algoritmo resultante toma el nombre de “*k-medias secuencial*”, reflejándose así que su característica principal es el uso de la variable tiempo t .

El proceso de predicción de los centroides mejora la eficiencia del “*k-medias*” básico, pero solo tiene sentido en aplicaciones como la aquí considerada, en la que el clasificador se ejecuta secuencialmente para organizar medidas (Y_{t-1}, Y_t, Y_{t+1}) con una relación temporal entre ellas:

$$Y_t = g(Y_{t-1}) \Rightarrow \bar{y}_{i,t} = g(\bar{y}_{i,t-1}) / i = 1:m \quad \langle 5.2 \rangle$$

En general, para cualquier par de sets de datos que muestre una evolución temporal de este tipo y una organización interna en un conjunto de k_t clusters ($\bigcup_{j=1}^{k_t} L_{j,t} \subseteq Y_t$), la evolución de cada dato $\bar{y}_{1:m,t}$ depende de la clase $G_{j,t} \equiv \{\bar{g}_{j,t}, L_{j,t}\}$ a la que éste esté asociada, por lo que el modelo que marque la evolución de cada $\bar{g}_{j,t} / j = 1:k$ será específico para cada clase.

En el caso de la tarea de MTT de interés, el set $Y_t = \{\bar{y}_{i,t} / i = 1:m\} / \bar{y}_{i,t} = [x_t \ y_t \ z_t]^T$ es un conjunto de medidas que representa la posición en t de un conjunto de k_t objetos en el espacio cartesiano tridimensional XYZ. Por lo tanto, aplicando el planteamiento anterior, la evolución del set de medidas Y_t , y por ende, de los $\bar{g}_{1:k,t}$ de los grupos incluidos en el mismo, vendrá dada por el modelo de movimiento de los k_t objetos en el espacio XYZ.

Existe, por tanto, una relación de equivalencia en lo que al modelo dinámico se refiere, entre cada $\bar{y}_{i,t} / i = 1:m_t$ y el centroide $\bar{g}_{j,t}$ de la clase $G_{j,t} \equiv \{\bar{g}_{j,t}, L_{j,t}\}$ a la que se asocia el dato ($y_{i,t} \in L_{j,t}$). Teniendo en cuenta este análisis, para realizar la predicción de los $\bar{g}_{1:k,t}$ se usa el mismo modelo CV planteado en el capítulo 2, que define la dinámica de los targets:

$$\bar{x}_{t+1} = \begin{bmatrix} x_{t+1} \\ y_{t+1} \\ z_{t+1} \\ vx_{t+1} \\ vz_{t+1} \end{bmatrix} = A \cdot \bar{x}_t + \bar{v}_t = \begin{bmatrix} 1 & 0 & 0 & t_s & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & t_s \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ z_t \\ vx_t \\ vz_t \end{bmatrix} + \begin{bmatrix} v_{x,t} \\ v_{y,t} \\ v_{z,t} \\ v_{vx,t} \\ v_{vz,t} \end{bmatrix}, \quad \langle 5.3 \rangle$$

$$\bar{y}_t = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = C \cdot \bar{x}_t + \bar{o}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ z_t \\ vx_t \\ vz_t \end{bmatrix} + \begin{bmatrix} o_{x,t} \\ o_{y,t} \\ o_{z,t} \end{bmatrix} \quad <5.4>$$

Para aplicar este modelo al proceso de predicción de centroides éste se modifica de modo que el vector de estado coincida con el de medidas ($\bar{x}_t = [x_t \ y_t \ z_t]^T = \bar{y}_t$), ya que es éste último el que está relacionado con el valor de los centroides a predecir ($\bar{y}_t \Rightarrow \bar{g}_{1:k,t}$). Así se establece una equivalencia directa entre el estado del modelo CV y los centroides ($\bar{x}_t \Rightarrow \bar{g}_{1:k,t}$).

Se reescriben, por tanto, las ecuaciones de estado y de salida para que el modelo CV modificado (\widehat{CV}) sea equivalente al CV :

$$\bar{x}_{t+1} = \begin{bmatrix} x_{t+1} \\ y_{t+1} \\ z_{t+1} \end{bmatrix} = A \cdot \bar{x}_t + B \cdot \bar{u}_t + \bar{v}_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} + \begin{bmatrix} t_s & 0 \\ 0 & 0 \\ 0 & t_s \end{bmatrix} \cdot \begin{bmatrix} vx_t \\ vz_t \end{bmatrix} + \begin{bmatrix} v_{x,t} \\ v_{y,t} \\ v_{z,t} \end{bmatrix}, \quad <5.5>$$

$$\bar{y}_t = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = C \cdot \bar{x}_t + \bar{o}_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} + \begin{bmatrix} o_{x,t} \\ o_{y,t} \\ o_{z,t} \end{bmatrix}, \quad <5.6>$$

donde el vector de entrada $\bar{u}_t = [vx_t \ vz_t]^T$ responde a la siguiente expresión:

$$\bar{u}_t = \begin{bmatrix} vx_t \\ vz_t \end{bmatrix} = \bar{u}_{t-1} + \bar{v}_{v,t-1} = \begin{bmatrix} vx_{t-1} \\ vz_{t-1} \end{bmatrix} + \begin{bmatrix} v_{vx,t-1} \\ v_{vz,t-1} \end{bmatrix} \quad <5.7>$$

Simplificando las expresiones <5.5> y <5.6> se obtiene la forma final del modelo \widehat{CV} , cuya descripción gráfica se muestra además en la Figura 5.10:

$$\begin{aligned} \bar{x}_{t+1} &= \bar{x}_t + B \cdot \bar{u}_t + \bar{v}_t \\ \bar{y}_t &= \bar{x}_t + \bar{o}_t \end{aligned} \quad <5.8>$$

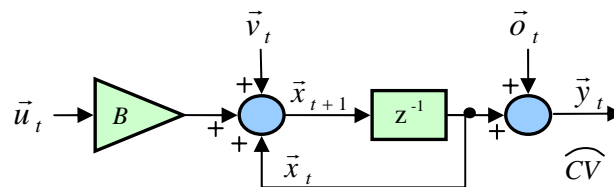


Figura 5.10. Diagrama de bloques del modelo CV modificado (\widehat{CV}) que se emplea en el proceso de predicción de cada centroide $\bar{g}_{j,t|t-1}$ de cada clase $G_{j,t} / j=1:k_t$.

A continuación se aplica al modelo \widehat{CV} la relación entre el $\bar{g}_{j,t}$ de cada clase y el vector de medidas \bar{y}_t . Al hacer esta equivalencia se tiene en cuenta que la segmentación es un

tratamiento determinístico del conjunto de medidas Y_t , y por tanto el ruido asociado a éstas \bar{o}_t es procesado por el segmentador e incluido en el valor de los $\bar{g}_{1:k,t}$. La relación resulta, por tanto, la que se muestra a continuación:

$$\bar{g}_{t,j} = \bar{y}_t + \bar{o}_t \quad \langle 5.9 \rangle$$

Aplicando la expresión $\langle 5.9 \rangle$ a la ecuación de medida en $\langle 5.8 \rangle$, ésta última queda de la siguiente forma:

$$\bar{y}_t = \bar{x}_t + \bar{o}_t / \bar{g}_{t,j} = \bar{y}_t + \bar{o}_t \Rightarrow \bar{g}_{t,j} = \bar{x}_t \quad \langle 5.10 \rangle$$

La conclusión que se extrae de la expresión anterior es que, usando el modelo \widehat{CV} , existe una equivalencia directa entre el $\bar{g}_{j,t}$ de cada clase $G_{j,t} / j=1:k_t$ y el vector de estado de este modelo de movimiento de cada objeto. Debido a ello, la ecuación de estado del modelo \widehat{CV} (en $\langle 5.8 \rangle$) permite predecir el valor del centroide de las clases $\bar{g}_{0,1:k,t} = \bar{g}_{1:k,t|t-1}$, resultando:

$$\bar{g}_{0,j,t+1} = \bar{g}_{j,t+1|t} = \bar{g}_{j,t} + B \cdot \bar{u}_{j,t} + \bar{v}_{j,t} / j=1:k_t, \quad \langle 5.11 \rangle$$

que al eliminar la componente de ruido que afecta a la predicción \bar{v}_t (pues ésta queda también procesada por el clasificador) puede expresarse como se muestra a continuación:

$$\bar{g}_{0,j,t} = \bar{g}_{j,t|t-1} = \bar{g}_{j,t-1} + B \cdot \bar{u}_{j,t-1} / j=1:k_t \quad \langle 5.12 \rangle$$

En la ecuación anterior, el vector de entrada $\bar{u}_{j,t-1}$ quedaría determinado por la característica de constancia típica del modelo \widehat{CV} (ver $\langle 5.3 \rangle$). Sin embargo, la aplicación de este modelo a la obtención del valor inicial de cada centroide $\bar{g}_{0,1:k,t} = \bar{g}_{1:k,t|t-1}$ permite definir un método de cálculo de $\bar{u}_{j,t-1}$ más elaborado y fiable. Para ello es necesario tener en cuenta que el valor que se genera para cada centroide $j=1:k_t$ a través de la ecuación $\langle 5.12 \rangle$ es una predicción ($\bar{g}_{j,t|t-1}$) que corrige el clustering para dar lugar a su valor final $\bar{g}_{j,t}$.

A partir de este razonamiento, la misma ecuación de estado del modelo \widehat{CV} define para cada clase $G_{j,t} / j=1:k_t$ un valor actualizado de $\bar{u}_{j,t}$, a usar en la siguiente ejecución en $t+1$, como se muestra gráficamente en la Figura 5.11 y analíticamente a continuación:

- Se extrae $\bar{u}_{j,t}$ de $\langle 5.12 \rangle$ y se sustituye el valor predicho $\bar{g}_{j,t|t-1}$ por el actualizado $\bar{g}_{j,t}$ del centroide de la clase j correspondiente:

$$\bar{g}_{j,t|t-1} = \bar{g}_{j,t-1} + B \cdot \bar{u}_{j,t-1} \Rightarrow \bar{u}_{j,t-1} = B^{-1}(\bar{g}_{j,t} - \bar{g}_{j,t-1}) / j=1:k_t \quad \langle 5.13 \rangle$$

- Se utiliza la constancia del modelo \widehat{CV} para propagar $\bar{u}_{j,t-1}$ de la expresión anterior:

$$\bar{u}_{j,t} = \bar{u}_{j,t-1} + \bar{v}_{v,j,t-1} \Rightarrow \bar{u}_{j,t} = B^{-1}(\bar{g}_{j,t} - \bar{g}_{j,t-1}) + \bar{v}_{v,j,t-1} = B^{-1}(\bar{g}_{j,t} - \bar{g}_{j,t-1}) / j=1:k_t \quad \langle 5.14 \rangle$$

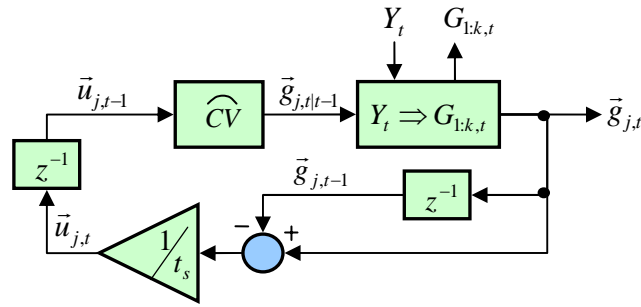


Figura 5.11. Diagrama de bloques del proceso de cálculo del vector de entrada $\bar{u}_{j,t-1}$ del modelo \widehat{CV} (Figura 5.10).

Este modelo se aplica a la predicción de valor del centroide de cada clase $\bar{g}_{j,t|t-1} / j = 1 : k_t$, por lo que el ruido de estado \bar{v}_t y medida \bar{o}_t se suponen incluidos en el proceso determinístico de clasificación $Y_t \Rightarrow G_{1:k,t}$.

- Finalmente, para resolver la ecuación <5.14> en la que aparece el cálculo de la inversa de la matriz singular B , se extraen las componentes de $\bar{u}_{j,t} = [vx_{j,t} \quad vz_{j,t}]^T$, y por ende las de $\bar{g}_{j,t} = [x_t \quad y_t \quad z_t]^T$:

$$\begin{aligned} vx_{j,t} &= (x_{j,t} - x_{j,t-1})/t_s \\ vz_{j,t} &= (z_{j,t} - z_{j,t-1})/t_s \end{aligned} \quad <5.15>$$

Las expresiones <5.12> y <5.15> resultantes son las utilizadas en la predicción del valor inicial de los centroides $\bar{g}_{0,1:k,t} = \bar{g}_{1:k,t|t-1}$, y se pueden reescribir de forma generalizada:

$$\bar{g}_{0,j,t} = \bar{g}_{j,t|t-1} = f_j(\bar{g}_{j,t-1}, \bar{u}_{j,t-1}) / j = 1 : k_t \quad <5.16>$$

$$\bar{u}_{j,t} = g_j(\bar{g}_{j,t}, \bar{g}_{j,t-1}) / j = 1 : k_t \quad <5.17>$$

donde $f_j()$ es la expresión general de la función de transición del modelo \widehat{CV} aplicado a la clase j , y $g_j()$ es una función genérica que representa la relación existente entre $\bar{u}_{j,t}$ y el valor actualizado de $\bar{g}_{j,t}$ de la clase j correspondiente en los instantes t y $t-1$.

La Figura 5.12 muestra el flujograma de funcionamiento del “*k-medias secuencial*”, en el que se ha añadido al del “*k-medias extendido*” de la Figura 5.5 (recuadrado con trazo azul) el proceso de cálculo del valor inicial de los centroides $\bar{g}_{0,1:k,t} = \bar{g}_{1:k,t|t-1}$, definido en su forma generalizada por las ecuaciones <5.16> y <5.17> (sobre fondo verde).

Como comentario final, hay que subrayar que para poder implementar el proceso de predicción de los centroides descrito es necesario mantener una identificación inequívoca a lo largo del tiempo de cada una de las clases. De otro modo la estimación de movimiento de éstos puede ser errónea, sobre todo en situaciones en que el valor de varios centroides coincide en el espacio de definición, o se crean nuevas clases en el proceso de clasificación.

La caracterización necesaria se consigue asignando a cada grupo un identificador desde el momento de su creación en la segmentación, de modo que su definición completa queda así:

$$G_{j,t} / j = 1 : k_t \equiv \{\bar{g}_{j,t}, L_{j,t}, \tau_j / j = 1 : k_t\} \quad \langle 5.18 \rangle$$

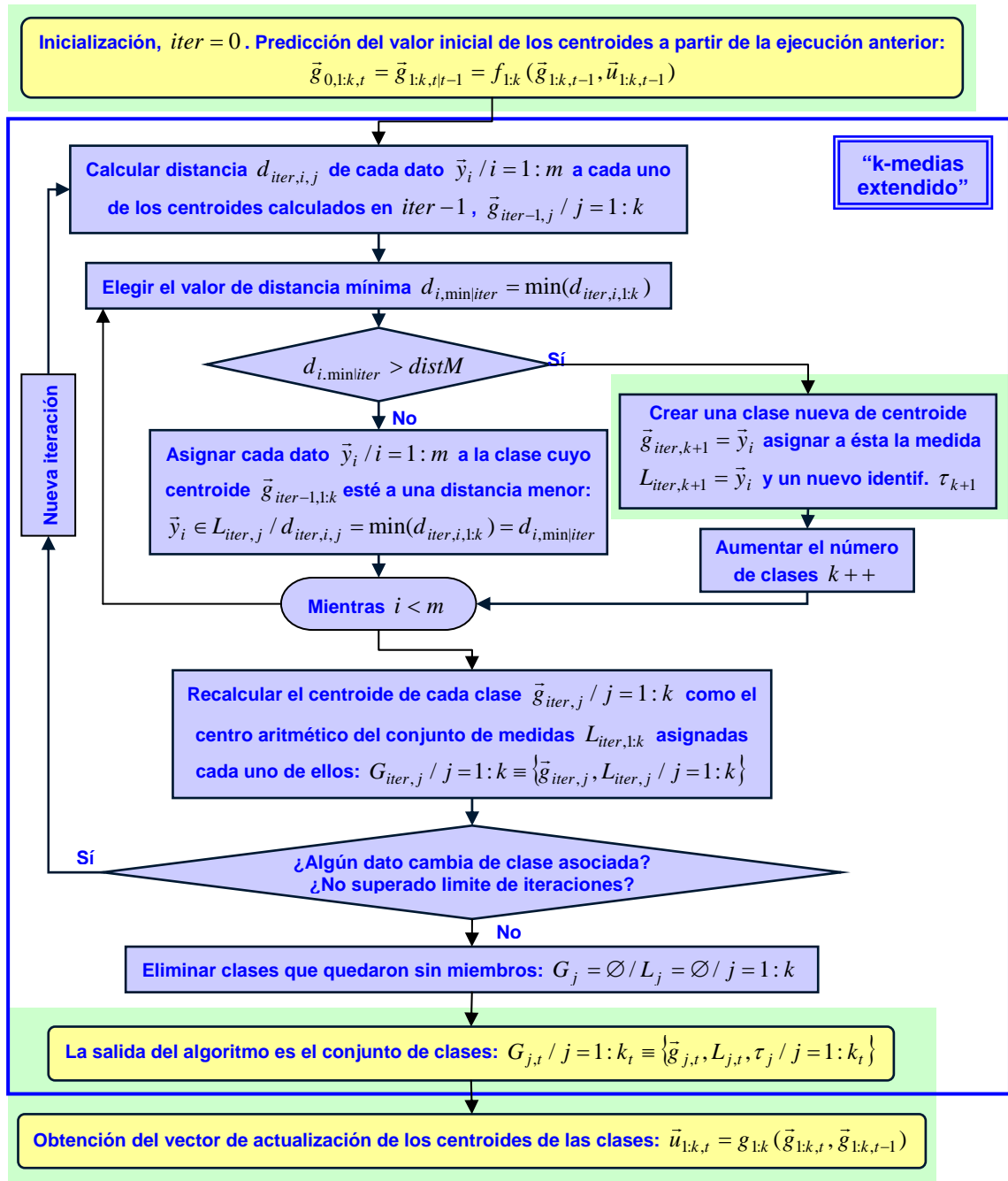


Figura 5.12. Diagrama funcional del “k-medias secuencial”. Las mejoras aportadas al flujograma del “k-medias extendido” de la Figura 5.5 se destacan con fondo verde.

En el caso concreto de la aplicación de interés, esta identificación implica que las medidas extraídas del objeto τ_j durante toda su trayectoria, se asignan al grupo $\tau_j : \{\bar{y}_{i,t} / i = 1 : l_j\} \in L_j$.

El identificador de grupo τ_j se convierte, por tanto, en un identificador de objetos, y más concretamente, del modelo de movimiento de cada uno de ellos.

Este análisis unido al carácter secuencial del “*k-medias secuencial*”, convierte al segmentador en un proceso de asociación, tal y como ya se expone en el capítulo 3. De hecho, la forma de agrupar datos del “*k-medias secuencial*” hace que éste se comporte como un proceso de asociación de tipo NN (“*Nearest Neighbour*”, “*Vecino Más Próximo*”, ver en el capítulo 2), de Y_t al conjunto de clusters $G_{1:k,t}$, y por ende, al conjunto de objetos implicados en la tarea de MTT, del que se extraen las medidas.

Bajo este planteamiento, el parámetro $distM$ que especifica la creación de un nuevo grupo en el “*k-medias secuencial*”, se comporta como el límite de gating del algoritmo NN. Las implicaciones de este parámetro comentadas anteriormente tendrán que ser tenidas en cuenta, por tanto, al utilizar el “*k-medias secuencial*” como algoritmo de asociación en la tarea de MTT.

Por su parte la Figura 5.13 muestra una simplificación del flujograma de la Figura 5.12. En la Figura 5.13 se aprecia más claramente que la propuesta de “*k-medias secuencial*” realizada en la tesis añade las etapas ya descritas al funcionamiento del “*k-medias extendido*”:

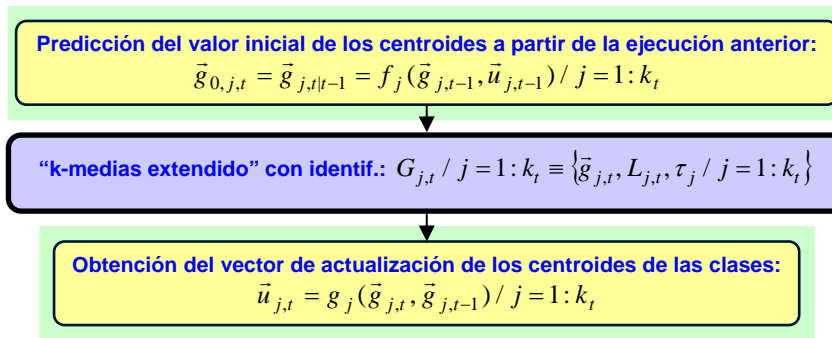


Figura 5.13. Flujograma simplificado del “*k-medias secuencial*” de la Figura 5.12.

1. La etapa previa de predicción de los centroides $\bar{g}_{0,1:k,t} = \bar{g}_{1:k,t|t-1} = f_{1:k}(\bar{g}_{1:k,t-1}, \bar{u}_{1:k,t-1})$,
2. la incorporación de un identificador τ_j a cada clase $G_{j,t} / j = 1:k_t$ generada por el “*k-medias extendido*”, asociado al valor inicial de cada centroide $\bar{g}_{0,1:k,t} \Rightarrow \tau_{1:k}$, y
3. una etapa posterior de actualización del vector de entrada $\bar{u}_{1:k,t} = g_{1:k}(\bar{g}_{1:k,t}, \bar{g}_{1:k,t-1})$, a usar en la de predicción de centroides de la ejecución siguiente del algoritmo $t + 1$.

Se puede concluir que el clasificador “*k-medias secuencial*” propuesto, se comporta como método de asociación de medidas a los distintos targets implicados en la tarea de seguimiento de múltiples objetos de interés en la tesis. La aplicación del clasificador como algoritmo de asociación en proceso de estimación multimodal basado en el PF se expone en el capítulo 6.

Finalmente, el hecho de que el proceso de predicción de centroides, y por tanto, el algoritmo de asociación propuesto, base su funcionamiento en un modelo de movimiento sencillo como el CV agiliza la ejecución del seguidor multi-objeto y no afecta a la fiabilidad del sistema de tracking global, tal y como se demuestra más adelante en este capítulo.

En cualquier caso, como se expone en apartados anteriores, es necesario añadir al flujograma del “*k-medias secuencial*” de la Figura 5.13 una etapa de validación de clases para que el segmentador cumpla las especificaciones de robustez y fiabilidad propuestas en la tesis.

5.2.5. Proceso de validación de clases

Tal y como se comenta en apartados anteriores, se ha diseñado un proceso de validación de clases que se añade al final del segmentador, con el objetivo de incrementar la robustez del clasificador frente al ruido y disminuir su sensibilidad frente a los parámetros implicados (*distM* en el caso del “*k-medias secuencial*”).

El sistema de validación permite, por tanto, mejorar la fiabilidad de la salida arrojada por el clasificador, y se aumenta la robustez de la estimación si se aplica el resultado de la segmentación a la frágil tarea de asociación incluida XPF, como se explica en el capítulo 3.

Al contrario que la mayoría de los algoritmos diseñados para la tesis, el proceso de validación aquí descrito está específicamente pensado para una aplicación de seguimiento como la de interés en este trabajo. Del mismo modo que ocurre con la etapa de predicción de los centroides que se añade al “*k-medias extendido*” para dar lugar al “*k-medias secuencial*”, este proceso aprovecha la ejecución secuencial de los clasificadores sobre conjuntos de datos consecutivos en el tiempo (Y_{t-1} , Y_t , Y_{t+1}) para tener un conocimiento más fiable y robusto sobre los grupos de datos implícitos en cada set Y_t .

A partir de esta premisa, el objetivo del sistema de validación es mantener una variable de validación $\beta_{j,t}$ asociada a cada grupo $G_{j,t} / j=1:k_t$ que informe de lo fiable que es la existencia y permanencia de este grupo en el set Y_t , en función de su aparición o no y verosimilitud en conjuntos de datos previos.

Conviene remarcar que en este punto se examina la funcionalidad del algoritmo de validación de forma genérica, para estudiar su aplicación a las dos alternativas de segmentación propuestas en esta tesis: el “*k-medias secuencial*” analizado en el apartado anterior, y el “*subtractive enlazado*” descrito posteriormente en este capítulo. Tras el análisis general, en este apartado se incluye también un análisis específico de su aplicación al “*k-medias secuencial*”, dejándose para más adelante el de su aplicación al “*subtractive enlazado*”.

a) Análisis funcional del proceso de validación de clases genérico

En la validación de cada cluster $G_{j,t} / j=1:k_t$, la validación se realiza en función de dos características de ésta:

1. Su proximidad $d_{t,j,jold}$ en el espacio de clasificación a la clase $G_{jold,t-1}$ más semejante de las obtenidas en el instante anterior $t-1$. Esta característica solo se puede alcanzar, como ya se ha comentado, en caso de que se ejecute secuencialmente el clasificador. De hecho, para definir la semejanza entre los dos conjuntos se utiliza el proceso de predicción de centroides que da lugar al “*k-medias secuencial*”. Es decir, la semejanza entre cada

conjunto $G_{j,t}$ y cada uno de los generados por el segmentador en $t-1$, $G_{1:k,t-1}$, se refiere a la distancia euclídea $d_{euclídea|j,jold}$ en el espacio de características usado, entre el centroide de la clase j ($\bar{g}_{j,t}$) y la predicción del de la clase $jold$ ($\bar{g}_{jold,t,t-1} / jold = 1:k_{t-1}$). Se elige el centroide para esta tarea por ser el parámetro del conjunto que se propaga, y la distancia euclídea $d_{euclídea|j,jold}$ por ser la medida de clasificación propuesta.

2. *Su verosimilitud*, $p_{j,t}$, $p_{j,1:t}$. La definición concreta de esta característica $p_{j,t}$ depende del clasificador al que se aplique el sistema de validación, si bien su función general es la de valorar la densidad de datos asociados a cada grupo $G_{j,t}$. Para suavizar el efecto de la variable $p_{j,t}$ en el proceso de validación, se usa un factor de olvido α_{valid} normalizado que pondera su valor en función de la evolución del grupo $G_{j,t}$ a lo largo del tiempo. El valor de $p_{j,1:t}$ se calcula, por tanto de la siguiente forma:

$$p_{j,1:t} = \alpha_{valid} \cdot p_{j,t} + (1 - \alpha_{valid}) \cdot p_{j,1:t-1} / j = 1:k_t \quad <5.19>$$

El objetivo de esta ponderación es el de incrementar la robustez del algoritmo de validación ante fluctuaciones instantáneas de la distribución que caracteriza a Y , muy comunes en la aplicación de seguimiento de interés, como se demuestra más adelante en la memoria.

Es evidente que para poder calcular la característica de verosimilitud $p_{j,1:t}$ de cada clase a través de la expresión <5.19> es necesario que éstas cuenten con un identificador τ_j que las relacione a lo largo del tiempo ($G_{jold,t-1} / jold = 1:k_{t-1} \Rightarrow G_{j,t} / j = 1:k_t$). En el caso del segmentador “*k-medias secuencial*” ya descrito, las clases ya cuentan con ese identificador τ_j a la salida del proceso, tal y como se expresa en la ecuación <5.18>. Esto no es así en el caso del “*subtractive enlazado*”, analizado en el apartado siguiente, por lo que en ese caso es necesario un proceso de identificación de las clases en el tiempo, como se explica más adelante.

A la vista de la relación entre la identificación de los clusters (o la asociación de tipo NN que éste implica) y el proceso de validación de las mismas, es evidente que este último además de validar su robustez valida también su correcta identificación, es decir la correcta asociación de las medidas a los targets en la aplicación de MTT. Este hecho es muy importante alcanzar la fiabilidad deseada en el seguidor multimodal, pues facilitar la eliminación de clusters espurios.

El proceso de validación diseñado actualiza linealmente el contador de validación $\beta_{j,t}$ asociado a cada grupo a partir de las dos características descritas, incrementándose en una unidad por cada condición de validación que cumpla el grupo a la salida del proceso de segmentación, y decrementándose en caso contrario.

La fiabilidad inicial de un conjunto j al ser creada es nula, por lo que su contador de validación se inicializa siempre con valor $\beta_{j,t} = 0$. Con este mismo criterio, los conjuntos cuyo contador de validación sea negativo a la salida del proceso son eliminados. Finalmente, el valor

positivo del contador se satura al alcanzar un límite $\beta_{j,t} = \beta_{valid}$, indicando que el conjunto es válido, es decir, la fiabilidad de su existencia es máxima.

Para decidir si el contador de validación $\beta_{j,t}$ de cada grupo ha de incrementarse o decrementarse se establece un límite para cada condición de validación (d_{valid} para la condición de proximidad dada por $d_{t,j,jold}$, y p_{valid} para la condición de verosimilitud dada por $p_{j,lt}$), además de un factor de histéresis normalizado, asociado al cada uno de ellos (π_d y π_p respectivamente). La Figura 5.14 muestra como queda organizado el espacio de las dos características de validación descritas tras la definición de estos límites.

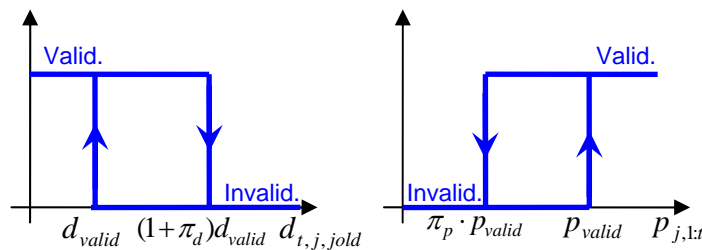


Figura 5.14. Representación del espacio de definición de las características $d_{j,jold}$ y $p_{j,lt}$, base del proceso de validación, en función de los correspondientes límites d_{valid} y p_{valid} , y factores de histéresis π_d y π_p .

La utilización de π_d y π_p permite definir, para cada una de las características implicadas, una zona de valor estable que no produce incremento ni decremento del contador de validación $\beta_{j,t}$. Los factores de histéresis comentados robustecen, por tanto, el funcionamiento de la etapa, insensibilizándolo del valor límite (d_{valid} y p_{valid}) de las dos condiciones de validación, tal y como se demuestra en apartados posteriores.

Por otro lado el límite de la condición de verosimilitud p_{valid} ha de definirse linealmente dependiente del número de clusters generados por el clasificador k_t , y saturado a un mínimo $\frac{p_{valid|max}}{k_{valid}}$ ³, ya que el valor de la verosimilitud de los clusters está normalizado $p_{l:k,lt} = 1$. La Figura 5.15 representa la función matemática que caracteriza el proceso de cálculo de p_{valid} en función de $p_{valid|max}$, k_{valid} y k_t .

El diagrama de funcionamiento del proceso de validación genérico, expuesto hasta este punto, se muestra en la Figura 5.16.

Gracias a la funcionalidad descrita, la validación añade dos campos al conjunto de características que proporciona el clasificador por cada grupo de salida:

³ En general k_{valid} suele fijarse suficientemente alto como para que la relación entre p_{valid} y k_t sea lineal para la mayor parte de las situaciones de clasificación.

$$G_{j,t} / j = 1 : k_t \equiv \{ \bar{g}_{j,t}, L_{j,t}, \tau_j, \beta_{j,t}, p_{j,t} / j = 1 : k_t \} \quad <5.20>$$

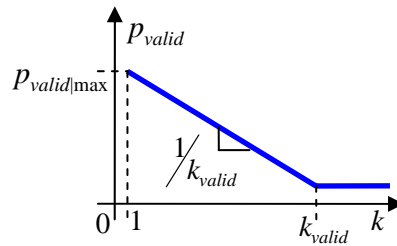


Figura 5.15. Representación gráfica de la forma de cálculo del valor límite de la característica de validación de verosimilitud p_{valid} en función del número de clases generadas por el clasificador k_t .

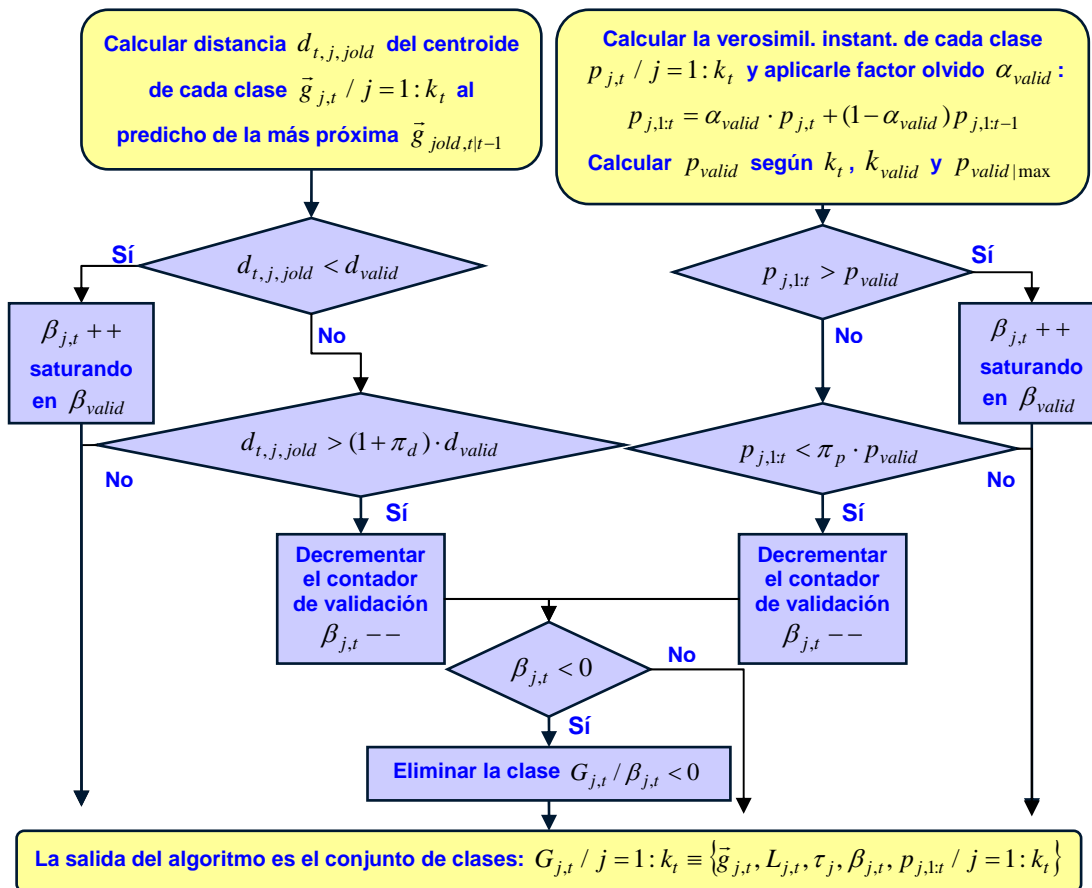


Figura 5.16. Diagrama explicativo de la funcionalidad general del proceso de validación de clases.

El valor de verosimilitud del grupo $p_{j,t}$ se incluye dentro del conjunto de características de cada grupo $G_{j,t} / j = 1 : k_t$ pues es utilizado por el tracker propuesto en la tesis (el XPFCP) en su paso de corrección, como se expone en el capítulo 6.

Por otra parte, la validación de las clases en las distintas tareas de clasificación implicadas en el XPFCP, establece una ventana temporal de admisión de las diferentes hipótesis de medida (y por ende de objetos implicados en la tarea de MTT de interés), modulada de forma directa a

través del parámetro β_{valid} , como se propone en [Blackmann99] (aplicado en ese caso a las técnicas MHT) y se propone en el trabajo descrito en [Ng06], analizado en el capítulo 2.

b) El proceso de validación de clases aplicado al “*k-medias secuencial*”

En el “*k-medias secuencial*”, la implementación de las condiciones de validación de cada clase $G_{j,t} / j=1:k_t$ es inmediata pues, como ya se ha comentado, este algoritmo incluye de forma implícita el proceso de identificación de las clases a lo largo del tiempo.

A continuación se especifica el valor de ambas condiciones para el “*k-medias secuencial*”:

1. Gracias al identificador τ_j , el cálculo de la distancia $d_{t,j,jold}$ se realiza únicamente respecto al centroide de la clase con su mismo identificador $\tau_{jold} = \tau_j$. Además, la predicción de los centroides ($\vec{g}_{jold,t|t-1} = \vec{g}_{0,j,t}$) producida en $t-1$ para ser usada como valor inicial de éstos en la clasificación en t se utiliza también en el proceso de validación para calcular la característica de proximidad $d_{t,j,jold}$.
2. Por otro lado, para asignar un valor de verosimilitud a cada grupo $G_{j,t} / j=1:k_t$ generado por un segmentador divisivo como el “*k-medias secuencial*”, lo más lógico parece hacerlo en función del número de datos asociados al éste $l_{j,t} / L_{j,t} = \left\{ \vec{y}_{i,j,t} / i=1:l_{j,t} \right\}$. Relativizando este valor en función del número de datos m_t del set $Y_t = \left\{ \vec{y}_{i,t} / i=1:m_t \right\}$, se obtiene el valor de verosimilitud $p_{j,t}$ buscado, a través de la siguiente expresión:

$$p_{j,t} = \frac{l_{j,t}}{m_t} / j=1:k_t \quad \langle 5.21 \rangle$$

A partir de esta definición, el valor de verosimilitud de la clase $p_{j,t}$ ponderado en el tiempo a través del factor de olvido α_{valid} se calcula como sigue:

$$p_{j,t} = \alpha_{valid} \cdot \frac{l_{j,t}}{m_t} + (1 - \alpha_{valid}) \cdot p_{j,t-1} / j=1:k_t \quad \langle 5.22 \rangle$$

Teniendo en cuenta la información de identificación que proporciona el “*k-medias secuencial*”, la aplicación a este clasificador del algoritmo de validación, no conlleva un incremento importante en su carga computacional, tal y como se demuestra en un apartado posterior, lo que facilita la consecución de la especificación de tiempo real perseguida.

El diagrama de funcionamiento del proceso de validación aplicado al “*k-medias secuencial*” se muestra en la Figura 5.17. Las únicas dos diferencias que aparecen en este flujograma respecto al de la Figura 5.16 son el cálculo de las características de proximidad y verosimilitud (marcado sobre fondo verde en la figura).

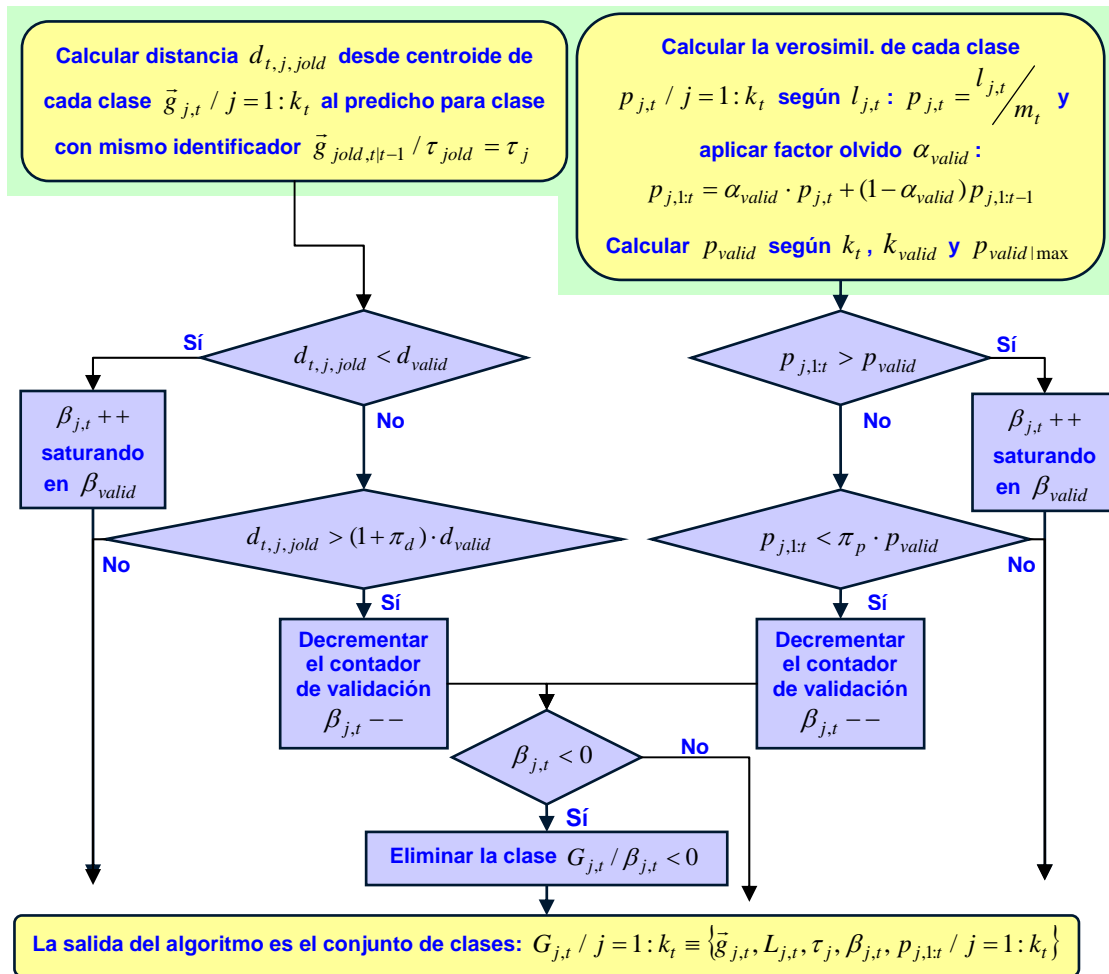


Figura 5.17. Diagrama funcional del proceso de validación aplicado al “k-medias secuencial”. Las etapas modificadas respecto a la versión genérica (Figura 5.16) se muestran con fondo verde.

Finalmente, en la Figura 5.18 se muestra el flujograma del “k-medias secuencial” con proceso de validación, que se obtiene al añadir al diagrama mostrado en la Figura 5.13 una llamada al de la Figura 5.17.

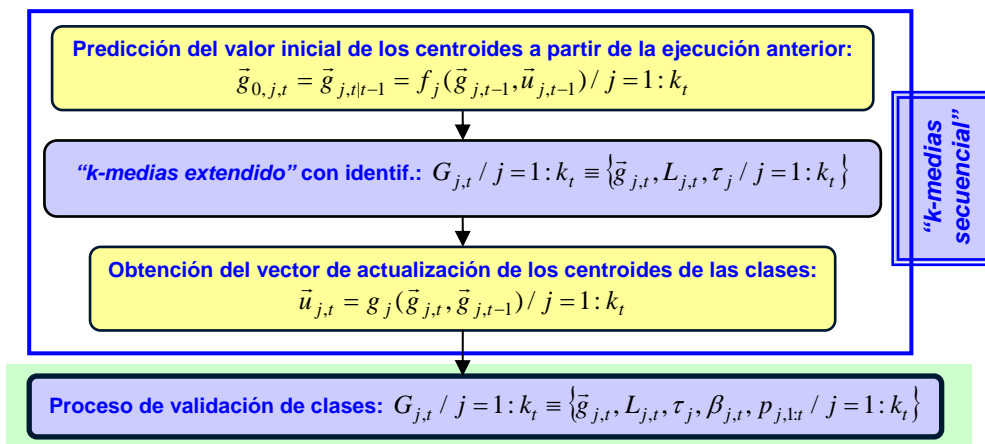


Figura 5.18. Diagrama funcional del “k-medias secuencial” con validación. La figura se obtiene al añadir al flujograma del “k-medias secuencial” (Figura 5.13) el proceso de validación de la Figura 5.17 (sobre fondo verde).

c) Análisis funcional del proceso de validación aplicado al “*k-medias secuencial*”

En este apartado se incluyen varios ejemplos que tienen por objeto ilustrar la funcionalidad general de la validación propuesta, y la específica de su aplicación al “*k-medias secuencial*”.

La Figura 5.19 muestra el resultado de la clasificación en un espacio bidimensional $\lambda_T = 2$, de cuatro sets de datos consecutivos en el tiempo, $Y_{t-1:t+2}$, generado por el “*k-medias secuencial*” sin proceso de validación. Cada grupo aparece representado a través de su centroide (cruz negra), el espacio limitado por el umbral de distancia euclídea $distM$ (círculo azul), y sus miembros (L_j), coloreados en función del grupo al que se asocian.

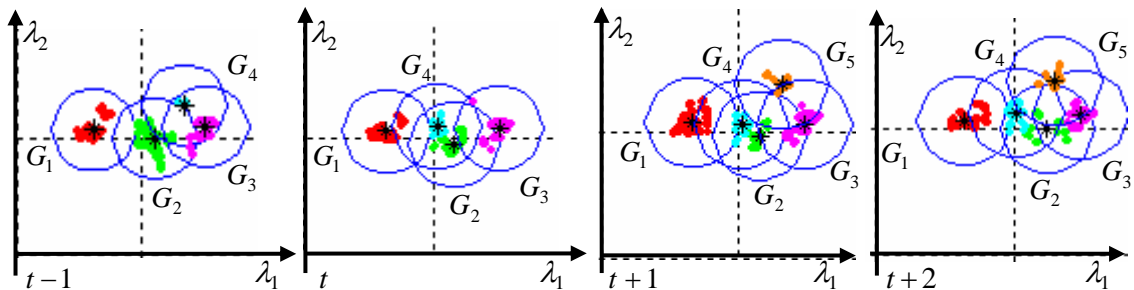


Figura 5.19. Ejemplo de ejecución del “*k-medias secuencial*” en un espacio bidimensional ($\lambda_T = 2$), frente a los sets consecutivos en el tiempo Y_{t-1} , Y_t , Y_{t+1} e Y_{t+2} (imágenes de izquierda a derecha). La evolución muestra una disminución de densidad de datos asociados a la clase G_4 , debido a lo cual ésta tiende confundirse con G_2

En esta Figura 5.19 se muestra un ejemplo típico de duplicación de clases, producida por el “*k-medias secuencial*”: debido a la inestable distribución de datos en el set a clasificar, los asociados a la clase $G_{4,t-1}$ (imagen izquierda) desaparecen casi en su totalidad en el instante t , y como consecuencia, a la clase $G_{4,t}$, mantenida por la ejecución secuencial del clasificador, se le asocian parte de los miembros que ocupan el espacio de $G_{2,t-1}$ (imagen a la derecha de la anterior). Se ha producido una duplicación de la clase $G_{2,t-1}$, por tanto, un error en su identidad.

En las dos imágenes de la derecha de la misma Figura 5.19 se muestra la salida del segmentador, de izquierda a derecha, en los dos instantes siguientes $t+1$ y $t+2$: en $t+1$ un conjunto de datos aparece en el espacio que en $t-1$ ocupaba la clase $G_{4,t-1}$, dando lugar a una nueva clase $G_{5,t+1}$, mientras que $G_{4,t+1}$ se mantiene duplicando a $G_{2,t+1}$. En $t+2$ la situación se estabiliza, concluyéndose que la pequeña alteración en la distribución de Y_t genera un error.

En la Figura 5.20 se presenta el resultado de segmentación de los mismos sets $Y_{t:t+2}$ mostrados en la Figura 5.19, generado en este caso por el “*k-medias secuencial*” con validación basada únicamente en la característica de distancia. En la imagen de la izquierda se observa que el grupo $G_{4,t}$ que duplica a la $G_{2,t}$ en la anterior Figura 5.19 no aparece en este caso. Este grupo es, de hecho, eliminado por el proceso de validación al no cumplir con la condición de

distancia impuesta por d_{valid} y π_d a su centroide. Sin embargo, la validación por distancia elimina el error de duplicación.

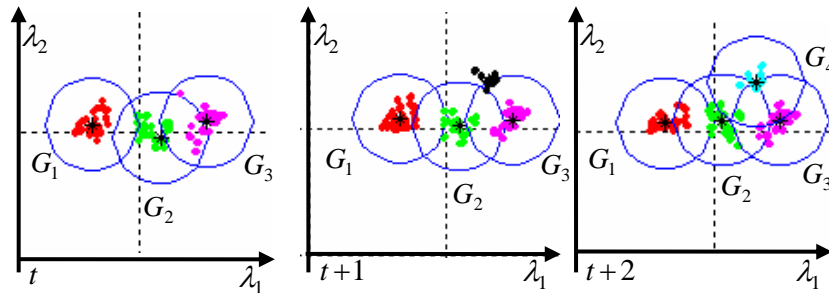


Figura 5.20. Ejemplo de ejecución del “*k-medias secuencial*” con validación basado únicamente en la condición de distancia, frente a los mismos sets $Y_{t:t+2}$ mostrados en la Figura 5.19.

Finalmente en la Figura 5.20 se muestra también como la nube de puntos que había sufrido una disminución de densidad en t vuelve a dar lugar al grupo desaparecido, pero no lo hace en $t+1$ (puntos de color negro en la imagen), sino en $t+2$. Esto se debe a que la clase no se valida hasta que su contador de validación $\beta_{j,t}$ no alcanza el valor fijado al parámetro β_{valid} , hecho que, al usar únicamente la condición de distancia, no ocurre hasta dicho momento. La validación por distancia produce, por tanto, un error de no generación de la clase $G_{4,t}$ en contrapartida al de duplicación que evita.

La Figura 5.21 muestra de nuevo la segmentación de los conjuntos $Y_{t:t+2}$, obtenida esta vez con el “*k-medias secuencial*” con el proceso de validación completo. En este caso, la condición de verosimilitud impuesta por p_{valid} , α_{valid} y π_p permite mantener la clase G_4 en todos los instantes de tiempo ($t-1:t+2$). Este hecho se debe a la funcionalidad del parámetro α_{valid} , que eleva la verosimilitud de la clase $p_{j,t}$ a pesar de los cambios de densidad del conjunto de puntos que se le asignan (reflejados en su verosimilitud instantánea $p_{j,t}$), debido a su valor histórico $p_{j,t-1}$. En este caso también el parámetro π_p tiene la función de insensibilizar la validación por verosimilitud del valor de ajuste de $p_{valid|max}$, que se fija para eliminar los outliers de la segmentación, tal y como demuestran los resultados de en apartados posteriores.

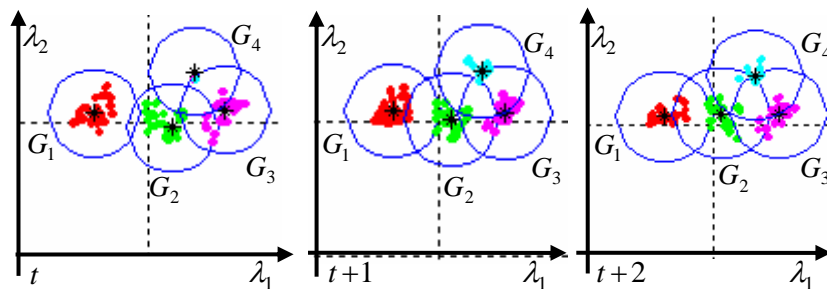


Figura 5.21. Ejemplo de ejecución del “*k-medias secuencial*” con el proceso de validación completo, frente a los mismos sets $Y_{t:t+2}$ mostrados en la Figura 5.19.

Con todo ello, se puede concluir que el proceso de validación completo mejora la robustez y la fiabilidad del “*k-medias secuencial*” eliminando los errores que genera el clasificador debidos, entre otras causas, a los continuos cambios en la densidad de las nubes de datos asociados a cada grupo. En el siguiente apartado se muestran otros efectos del proceso de validación en la funcionalidad del clasificador finalmente obtenido: la fiabilidad del proceso de identificación, y el robustecimiento del “*k-medias secuencial*” frente al ruido y a modificaciones de todos sus parámetros.

5.2.6. Resultados comparativos del “*k-medias secuencial*” con proceso de validación respecto al “*k-medias extendido*”

Para examinar el comportamiento del clasificador propuesto, se comparan los resultados que arroja con los generados por el “*k-medias extendido*”, en los experimentos descritos en el apéndice III, y se observan distintas variables de exactitud, tiempo y robustez.

a) Tasa de error:

La Figura 5.22 muestra la tasa de error del experimento complejo (a) y del completo corto (b) por los algoritmos “*k-medias extendido*” (en tonos azules) y “*k-medias secuencial*” (en tonos rojos) sin validación. A la vista de la esta figura se extraen las siguientes conclusiones:

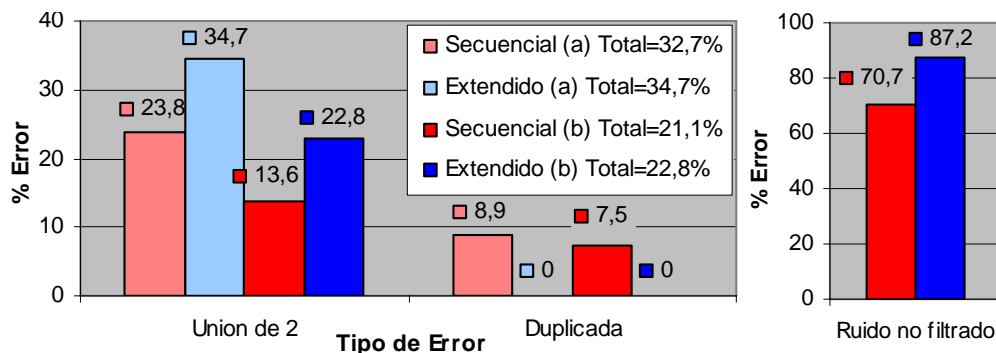


Figura 5.22. Tasa de error del “*k-medias secuencial*” sin proceso de validación (en tonos rojos) y del “*k-medias extendido*” (en tonos azules) en los experimentos complejo (a) y completo corto (b).

- La tasa de error total del segmentador propuesto es algo menor a la del “*k-medias extendido*” en ambos casos (32.7% frente a 34.7% en el experimento complejo, y 21.1% frente a 22.8% en el experimento completo corto), aún sin validación.
- Haciendo un análisis más profundo se observa que el clasificador propuesto genera muchos menos errores de unión de 2 clases en 1, de los generados por el “*k-medias extendido*”. Este resultado es sobre todo evidente en el segundo tercio del experimento complejo (marcado como (a) en la Figura 5.22), en el que hay 3 personas muy juntas en la escena. La mejora que aporta el algoritmo propuesto en este tipo de situaciones se debe al proceso de predicción de centroides, que permite que las clases permanezcan a lo largo del tiempo, incluso cuando sus miembros son muy semejantes, como ya se había comentado en el apartado anterior. La robustez que aporta el “*k-medias secuencial*” a la aplicación de seguimiento ante situaciones en las que los objetos se encuentran muy juntos queda, de este modo, probada.

- En contrapartida a lo anterior, el segmentador propuesto produce un error de duplicación de grupos no producido por el “*k-medias extendido*”. Este error aparece al final del experimento complejo (situación (a) en la Figura 5.22) cuando una de las personas en escena es ocultada casi totalmente por otra. En este tipo de situaciones, la pobreza del conjunto de medidas extraídas del target, junto a la inexactitud del modelo de predicción provoca que el valor predicho del $\bar{g}_{1:k,t}$ no sea el adecuado, y que se aparezca un error de duplicación, como se explica en el apartado anterior, a través de la Figura 5.19 y la Figura 5.20 anteriores. Es importante recordar, que gran parte de estos errores, al margen de ser menos relevantes en la aplicación de interés que los de unión de grupos, son eliminados por la validación.
- En lo que al ruido se refiere, el comportamiento de ambos algoritmos es bastante deficiente: el “*k-medias secuencial*” filtra el 29.3% de los ruidos, mientras que el “*k-medias extendido*” lo hace en el 12.8% de las ocasiones, al ser ejecutados en el experimento de ruido. La mejora que aporta la propuesta diseñada se debe a que si el ruido se encuentra próximo a un objeto ya existente, los outliers se asocian a éste, por lo que no se crea un grupo nuevo. No se trata, por tanto, de una mejora real en el comportamiento frente al ruido del clasificador.

Teniendo en cuenta todo lo anterior se concluye que el segmentador propuesto presenta, en cualquier caso, una mayor fiabilidad y robustez que el “*k-medias extendido*”.

Por su parte, la Figura 5.23 muestra la comparativa de tasa de error del “*k-medias secuencial*” sin (en tonos rojos) y con (en tonos amarillos) validación en los mismos experimentos, a la vista de la cual se extraen nuevas conclusiones:

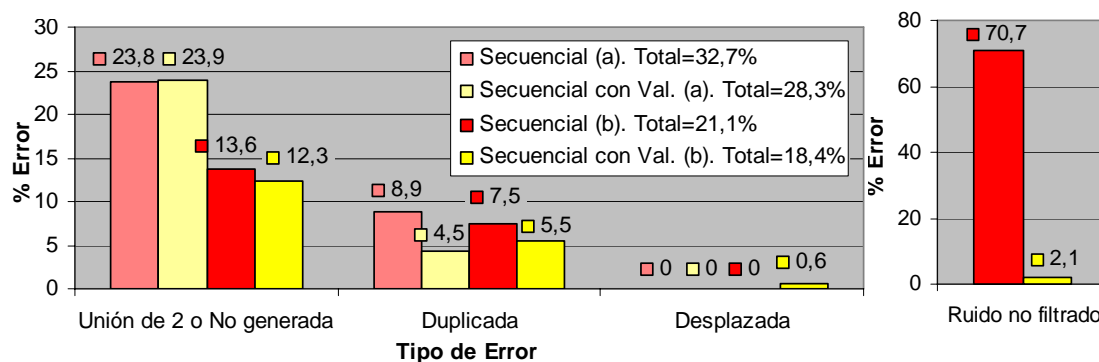


Figura 5.23. Tasa de error del “*k-medias secuencial*” sin validación (en tonos rojos) y con validación (en tonos amarillos) en los experimentos complejo (a) y completo corto (b).

- El proceso de validación disminuye sobre todo la tasa de error de duplicación que arroja el “*k-medias secuencial*” tal y como se explica en el apartado anterior, producido de forma indirecta por la etapa de predicción de centroides. La reducción es del 50% en situaciones complicadas como las que aparecen en el experimento complejo, en las que los targets de los que se obtiene Y_t están parcialmente ocultos en muchas ocasiones.
- El número de errores de unión de 2 clases en 1 o de no generación de una clase se mantiene más o menos constante con la incorporación de la validación al “*k-medias secuencial*”. La

baja densidad de puntos asociados a algunas clases, junto a la inexactitud en la predicción de su centroide, hace que esta tasa de error sea difícil de mejorar en situaciones complejas, con la validación más de lo que lo hace el “*k-medias secuencial*”.

- En general la tasa global de error del “*k-medias secuencial*” disminuye poco al añadir la validación (un 3.5% más o menos, respecto a la tasa del clasificador sin validación).
- Sin embargo, el rechazo al ruido del algoritmo aumenta considerablemente al añadir la validación al segmentador, pues la tasa de ruido filtrado aumenta del 29.3% al 97.9%, en el experimento de ruido. La eficiencia de la validación queda, por tanto, comprobada. Es interesante destacar el hecho de que la introducción de la condición de verosimilitud en la validación aumenta la tasa de rechazo de ruido del “*k-medias secuencial*” del 61.2% al 97.9% final. Este último dato confirma de nuevo la necesidad de uso de las dos características de validación en el proceso, pues la incorporación de cada una de ellas incrementa en más de un 30% su rechazo al ruido.

Con todo lo anterior, queda patente que el proceso de validación mejora la fiabilidad del clasificador, sobre todo en lo referente a su robustez al ruido, uno de los objetivos principales de su implementación, como se expone en apartados anteriores.

La otra funcionalidad principal de la validación, es la de disminuir la sensibilidad del segmentador frente a variaciones del parámetro $distM$. La Figura 5.24 muestra la fiabilidad del “*k-medias secuencial*” con validación frente a variaciones de $distM$ en el test complejo.

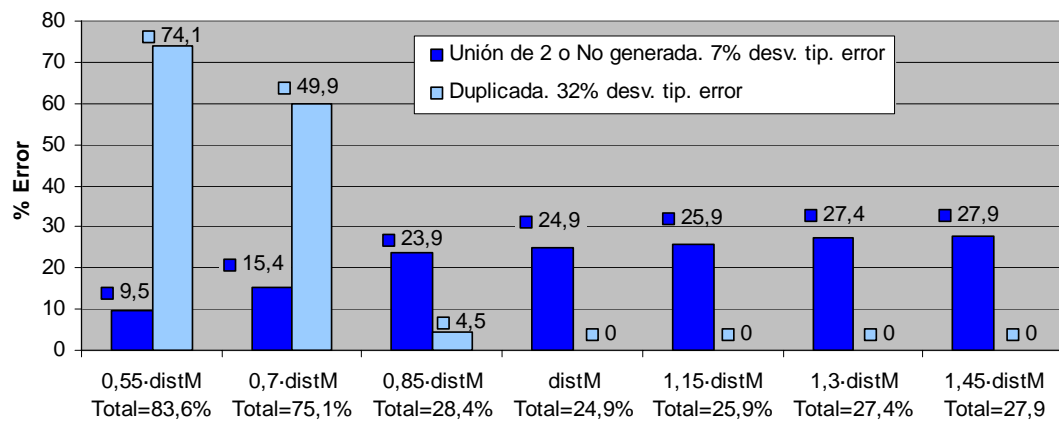


Figura 5.24. Tasa de error del “*k-medias secuencial*” con proceso de validación en el experimento complejo en función del valor de $distM$.

Tal y como se aprecia en la Figura 5.24, la tasa de error permanece prácticamente constante (nula en lo que a errores de duplicación se refiere) a partir de un valor determinado del parámetro (dentro de un margen lógico). Este resultado es mucho mejor que el que arroja el “*k-medias extendido*” ante el mismo experimento (ver la Figura 5.9), en el que su fiabilidad demuestra ser muy dependiente del valor de $distM$ (la tasa de error no se estabiliza para ningún valor del parámetro). La menor sensibilidad del “*k-medias secuencial*” con validación permite que la elección del parámetro no sea tan crítica, y que, por tanto, el sistema sea más flexible al tamaño y forma de las clases, lo cual es uno de los objetivos perseguidos en esta tesis.

Teniendo en cuenta que la validación introduce siete parámetros nuevos (β_{valid} , α_{valid} , $p_{valid|max}$, k_{valid} , π_p , π_d y d_{valid}), cuyo efecto en la funcionalidad del clasificador ya ha sido comentado, es necesario hacer un análisis de sensibilidad del algoritmo respecto a su valor.

La Figura 5.25 muestra la desviación respecto a la media de error del “*k-medias secuencial*” con validación, ante la variación de 0 a 1 de los cuatro parámetros normalizados (α_{valid} , $p_{valid|max}$, π_p , π_d) en el experimento complejo. A la vista de los resultados mostrados, es evidente que la validación no es excesivamente sensible al correcto ajuste de los parámetros mencionados (el valor medio de las desviaciones típicas es del 3.3%), lo cual es importante a la hora de cumplir la especificación de robustez establecida en la tesis.

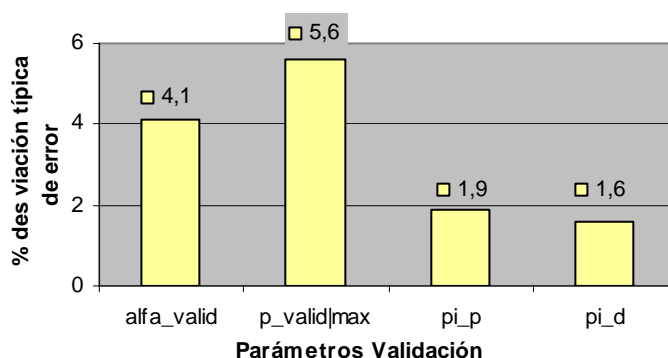


Figura 5.25. Desviación respecto a la media de error del “*k-medias secuencial*” con validación en el experimento complejo para los parámetros: α_{valid} (*alfa_valid*), $p_{valid|max}$ (*p_valid|max*), π_p (*pi_p*) y π_d (*pi_d*).

El algoritmo de validación es también poco sensible al parámetro β_{valid} , pues su modificación de 1 a 5 en el mismo experimento produce una desviación sobre la media de la tasa de error total del 1.5%. Su correcto ajuste no tiene por tanto tampoco una repercusión muy importante, siempre que se mantenga por encima de cero, para se ejecute el proceso de validación. Además, es necesario tener en cuenta que el valor de β_{valid} está relacionado con el número de validaciones o invalidaciones secuenciales (en función de las dos condiciones explicadas) que ha de experimentar un grupo para que se valide o elimine respectivamente.

En lo que al ajuste del parámetro d_{valid} se refiere, cabe destacar que para que la validación por distancia se realice de forma adecuada, su valor ha de fijarse en función de la velocidad máxima que se espera en el movimiento de las clases de una ejecución a otra del segmentador⁴. Si se usa un valor para d_{valid} mayor, el clasificador puede confundir las clases cuando estas crucen su posición en el espacio elegido, y si se usa uno menor la validación no será flexible a los cambios bruscos de dinámica de las clases, en contra de las especificaciones de la tesis.

Finalmente se analiza la tasa de error del “*k-medias secuencial*” con y sin validación en lo que a su tarea como identificador de objetos se refiere. Con este fin se compara de forma manual el identificador que el “*k-medias secuencial*”, sin y con validación, le asigna a cada

⁴ Esta especificación llevada a la aplicación de interés se refiere a la máxima velocidad a la que pueda moverse un objeto en entornos interiores y depende además del periodo de muestreo del sistema de medida.

target del entorno con la identidad real de cada uno ellos. En esta comparativa no se incluyen los resultados arrojados por el “*k-medias extendido*” pues éste no incorpora identificación.

La Figura 5.26 muestra los resultados de identificación obtenidos ejecutando el “*k-medias secuencial*” sin y con validación (imagen izquierda y derecha respectivamente) en la segunda parte del experimento complejo. En esta figura se representa por colores en función de su identidad, la posición en el plano XZ del $\bar{g}_{1,5}$ asociado a cada uno de los targets en la escena a lo largo del tiempo. Sobre esta secuencia de puntos de color se muestra con una flecha el “*ground truth*”⁵ manual con el que comparan los resultados de identificación arrojados por los clasificadores, indicando con ella además la dirección del movimiento de los targets. Varias conclusiones se pueden extraer a la vista de esta figura:

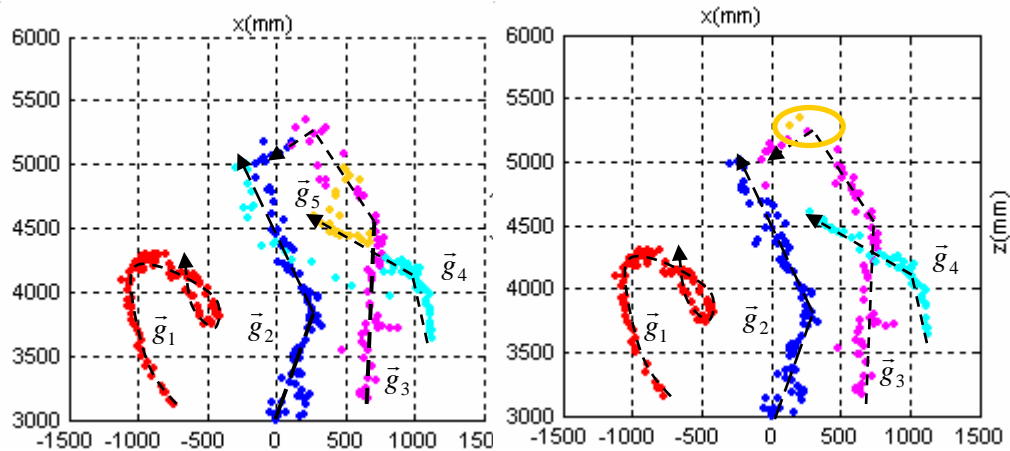


Figura 5.26. Representación por colores de la posición de $\bar{g}_{1,5}$, proyectada en el plano XZ , de cada uno de los objetos identificados por el “*k-medias secuencial*” sin (imagen izquierda) y con validación (imagen de la derecha). Los resultados se han obtenido ejecutando los algoritmos en la segunda parte del experimento complejo.

- La tasa de error del algoritmo sin validación es del 8.5%, pues se producen las comentadas confusiones de identidad, debidas a la oclusión total del grupo G_4 por el G_3 (el objeto asociado a G_4 cuyo centroide se marca en color azul claro en la Figura 5.26, pasa antes por el punto de cruce que el asociado a G_3 , de centroide representado en magenta en la figura). Debido a ello se crea la clase G_5 , cuyo centroide (de color naranja en la imagen izquierda de la figura) sigue la trayectoria del objeto asociado inicialmente a G_4 . Por su parte, la clase G_4 pasa a asociarse al mismo objeto que se encuentra asociada G_2 desde el principio del experimento, generando así además un error de duplicación. La identificación de la clase G_4 es así errónea en el 65% de las ejecuciones del test (un 4.3% del 8.5% total).
- El problema comentado en el párrafo anterior no lo produce el segmentador con validación (imagen de la derecha en la Figura 5.26), que únicamente genera una clase duplicada de G_3 en dos instantes del test (dos puntos de color naranja marcados por un círculo en la imagen)

⁵ El término inglés “*ground truth*” se utiliza recursivamente en la literatura científica para referirse al resultado verdadero o de referencia de un algoritmo, por lo que se utiliza sin traducir en el documento.

que desaparece rápidamente. La tasa de error global en este caso es del 0.6%, es decir, que la tarea de identificación se realiza correctamente.

- El objeto asociado a la clase G_1 es perfectamente identificado a lo largo de todo su recorrido que incluye cambios bruscos de dirección, por los dos clasificadores (con validación en la imagen izquierda y sin validación en la derecha de la Figura 5.26). Este hecho demuestra la ya comentada fiabilidad del proceso de identificación implícito en el “*k-medias secuencial*” a pesar de la simplicidad del modelo de predicción de centroides usado.

Este análisis permite afirmar que el “*k-medias secuencial*” diseñado realiza la tarea de identificación prácticamente sin error al añadirle el proceso de validación propuesto.

La Figura 5.27 presenta imágenes extraídas de la parte del experimento complejo en las que se aprecia la situación de oclusión comentada. En las imágenes se muestra el resultado generado por de los algoritmos, de izquierda a derecha, “*k-medias extendido*”, “*k-medias secuencial*” sin y con validación, ante la misma situación real. Los dibujos inferiores muestran la proyección de los resultados sobre el plano del suelo (XZ), y los superiores sobre la imagen obtenida con el sistema de sensado usado en la aplicación de MTT. El círculo que representa a cada clase generada por los segmentadores en XZ , y que queda como un cilindro al prolongarse en la coordenada de altura (y), se proyecta como un rectángulo en la imagen (dibujos superiores).

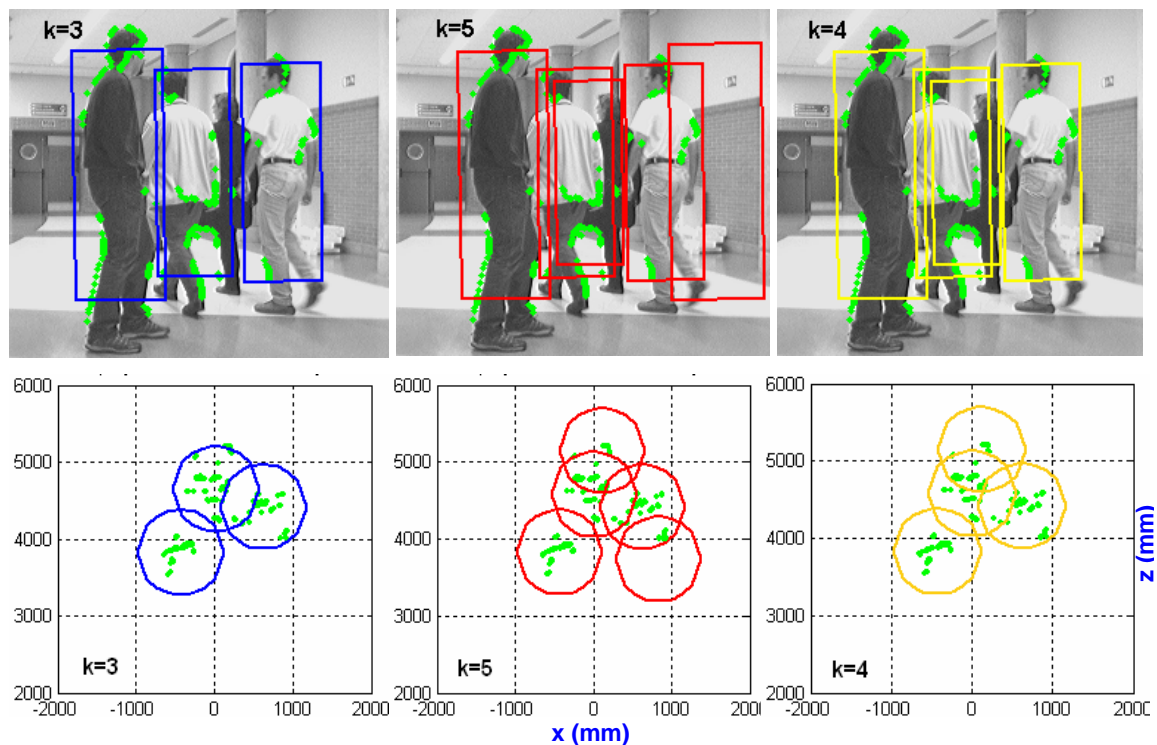


Figura 5.27. Resultados reales comparativos obtenidos del experimento complejo. En las figuras superiores se muestra su proyección sobre la imagen, y en las inferiores sobre el plano del suelo (XZ). Además, en ambas proyecciones se representa en verde la proyección de los datos de Y .

La Figura 5.27 confirma las conclusiones más importantes extraídas de las figuras anteriores: el “*k-medias extendido*” produce error de unión de 2 clases en 1 cuando los objetos de los que

se extraen los datos están muy próximos o superpuestos, como ocurre con la persona del fondo en la imagen de la izquierda; el “*k-medias secuencial*” resuelve este problema gracias al paso de predicción de centroides, pero a cambio genera duplicación de clases, cuando éstas tienen pocas medidas asociadas (imagen del centro); y el proceso de validación añadido a éste último mejora el comportamiento final del clasificador, eliminando las clases duplicadas y generando una sola clase correctamente identificada por cada objeto en la escena (imagen de la derecha).

La suma de todos los análisis expuestos, permite concluir que el “*k-medias secuencial*” con validación alcanza las especificaciones de flexibilidad, fiabilidad y robustez establecidas.

b) Tiempo de ejecución y número de iteraciones:

Una de las mejoras más relevantes que aporta el “*k-medias secuencial*” respecto a la versión básica del clasificador es la disminución de su t_{exe} , como se comenta en varios apartados anteriores, y se demuestra en este punto.

En la Figura 5.28 se incluyen las gráficas comparativas del t_{exe} ⁶ (imagen superior) y el número de iteraciones (imagen inferior) empleadas por el “*k-medias extendido*” (en azul) y el “*k-medias secuencial*” con validación (en rojo), al ser ejecutados en el experimento complejo. Por otro lado, en la Figura 5.29 se muestra la media del t_{exe} (imagen derecha) y del número de veces que iteran (imagen izquierda) los mismos clasificadores, al ejecutarlos en el experimento completo largo, en función del número de clases generadas por éstos k_t .

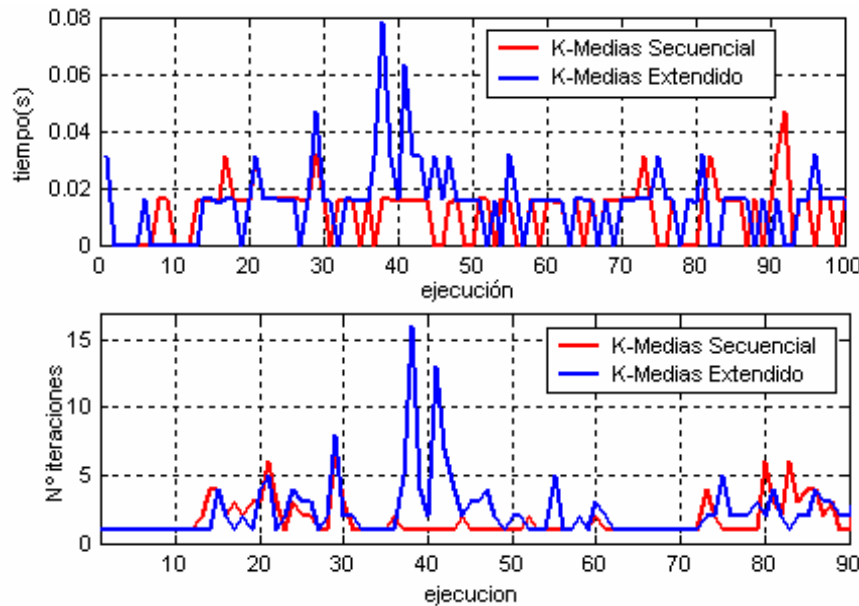


Figura 5.28. Gráficas comparativas del t_{exe} y el número de iteraciones de los algoritmos “*k-medias extendido*” y “*k-medias secuencial*” con validación, obtenidas en el experimento complejo.

⁶ Las medidas de t_{exe} mostradas en este capítulo se realizan sobre una plataforma de pruebas desarrollada en entorno Matlab, y corriendo en un procesador Intel Centrino DUO a 1.8GHz.

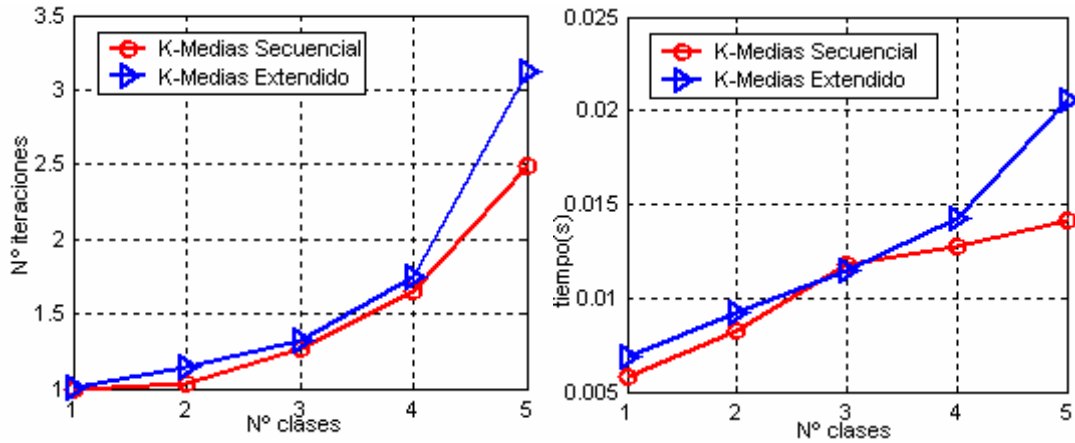


Figura 5.29. Gráficas comparativas del t_{exe} (imagen izquierda) y la media de veces que iteran (imagen derecha) los algoritmos “*k-medias secuencial*” con validación y “*k-medias extendido*”, en función del número de clases que arrojan como salida k_t en el experimento completo largo.

No se incluye en las figuras comparativas (Figura 5.28 y Figura 5.29) los resultados generados por el “*k-medias secuencial*” sin validación, puesto que, como se espera, no se encuentra una diferencia apreciable entre el t_{exe} de éste y el del “*k-medias secuencial*” con validación. De las pruebas temporales se extraen las siguientes conclusiones:

- Tanto el t_{exe} , como el número de veces que itera la versión secuencial del “*k-medias*” con validación, son menores que los de la versión básica (Figura 5.28).
- El t_{exe} de ambos procesos está relacionado directamente con el número de veces que iteran. De hecho, tal y como se aprecia en la Figura 5.28, el t_{exe} del “*k-medias extendido*” sólo sufre un incremento apreciable si el segmentador itera un número de veces mayor al medio.
- La mejora en velocidad de ejecución del segmentador propuesto es sobre todo evidente en situaciones complejas de clasificación. En la Figura 5.28, los máximos de la gráfica azul, relacionada con el “*k-medias extendido*” y que no aparecen en la gráfica roja (asociada al “*k-medias secuencial*” con validación), coinciden con la aparición de un nuevo objeto en la escena que produce la oclusión parcial de otro y complica la segmentación (situación que coincide con la escena mostrada en la Figura 5.27).
- Por otro lado, el número de veces que itera el algoritmo (imagen izquierda de la Figura 5.29), y por tanto su t_{exe} (imagen derecha), depende de k_t . Esta relación aparece en cualquier proceso de clasificación, si bien en el caso del “*k-medias extendido*” la dependencia pasa a ser exponencial en situaciones complejas (conforme aumenta k_t), cuestión que no ocurre con el “*k-medias secuencial*” propuesto.
- Finalmente, se ha comprobado además que número de iteraciones que se ejecutan ambos segmentadores, y por ende su t_{exe} , no depende de la cantidad de datos a organizar m_t . Este hecho se analiza en mayor detalle al final de este capítulo.

Como resumen, se concluye que, sobre todo en situaciones complejas, el “*k-medias secuencial*” propuesto en la tesis tiene un menor t_{exe} que el “*k-medias extendido*”, incluso añadiéndole el proceso de validación, debido a que itera menos veces.

c) Distorsión y separación de las clases:

Calculando los parámetros de distorsión y separación de las clases, obtenidos en el experimento complejo con los algoritmos “*k-medias secuencial*” con (en amarillo) y sin validación (en rojo) y “*k-medias extendido*” (en azul), se extraen los resultados mostrados en la Figura 5.30. El valor de distorsión arrojado por el “*k-medias secuencial*” con validación no se presenta en la figura, pues es similar al generado por el clasificador sin validación.

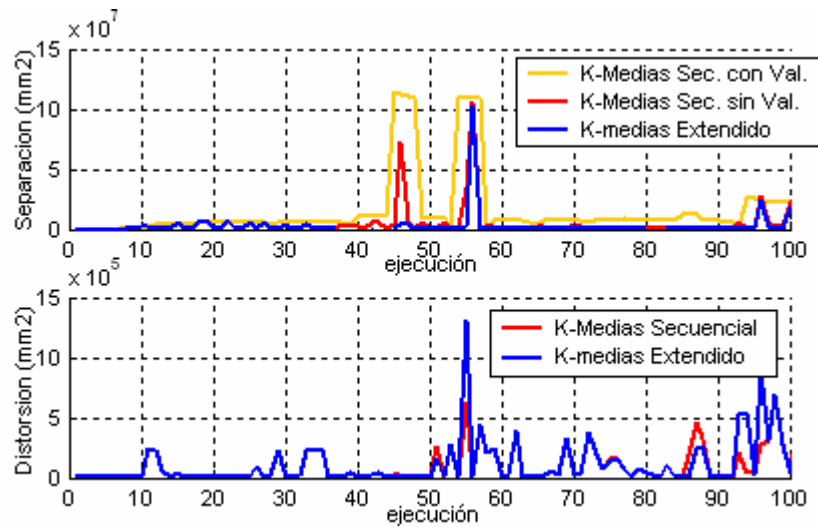


Figura 5.30. Gráficas comparativas de los valores de separación y distorsión que arrojan los algoritmos “*k-medias secuencial*” y “*k-medias extendido*”, al ser ejecutados en el experimento complejo.

Las formas de onda mostradas en la Figura 5.30 son parecidas a las de la Figura 5.28, pues el valor que adoptan estos factores de calidad responde al mismo comportamiento que el del t_{exe} : los segmentadores arrojan valores similares de separación y distorsión en todo el experimento, excepto en las situaciones más complejas.

De hecho, el momento de mayor diferencia aparece en el mismo instante del experimento comentado anteriormente, en el que aparece una cuarta persona en la escena (alrededor de la ejecución 55 del experimento). En ese punto, el “*k-medias secuencial*” con validación presenta un valor de separación mucho mayor que el que muestra el mismo clasificador sin validación y el “*k-medias extendido*”. Este hecho implica que el segmentador propuesto realiza correctamente la asignación de medidas a la nueva clase, mientras que los otros no lo hacen, produciendo errores de no generación de una clase (el “*k-medias extendido*”) o de duplicación (el “*k-medias secuencial*” sin validación), como se demuestra en apartados anteriores.

Los resultados extraídos de este último análisis vuelven, por tanto, a corroborar la efectividad del “*k-medias secuencial*” con validación, propuesto en la tesis.

5.3. El algoritmo “*subtractive enlazado*”

El algoritmo “*subtractive*” ([Chiu94] o [Dubois97]), en su versión básica, es una extensión del “*método mountain*” de clasificación ([Yager94]). Se trata de un clasificador borroso, recursivo y repetible, y puede ser usado para organizar un conjunto Y en un número de grupos k no fijado previamente.

La organización que realiza el “*subtractive*” se basa en la suposición de que cada punto i del set a ordenar (\bar{y}_i) es el centroide potencial de un grupo j (\bar{g}_j) con una probabilidad p_i que depende de la densidad de puntos de Y que lo rodean en el espacio de características del clustering. El segmentador se basa, por tanto, en el análisis de la PDF que implica la distribución del set Y en el espacio de características.

El carácter borroso del segmentador se consigue a través del valor de probabilidad que se calcula para cada punto i del set (p_i), pues puede utilizarse para definir el nivel de pertenencia del éste a cada clase j . De este modo se borrosifica la asociación de cada \bar{y}_i con cada clase, permitiéndose que pertenezca a ninguna, una o varias de las clases extraídas por el clasificador.

Es importante destacar que en la especificación básica del algoritmo no se hace uso de esta característica, pues no se define la asignación de cada \bar{y}_i con alguna de los k grupos generados a la salida. Se trata, por tanto, de un proceso no enlazado, si bien en muchas de sus aplicaciones suele usarse una modificación de carácter enlazado y borroso, como la que aquí se explica.

Tal y como ya se ha comentado en apartados anteriores, la condición borrosa del “*subtractive*” permite obtener una segmentación más robusta frente a la existencia de outliers en el conjunto Y , que la que se consigue con algoritmos booleanos como el “*k-medias*”.

Por otro lado, el clustering borroso que genera el “*subtractive*” es especialmente interesante en la aplicación de MTT objetivo de la tesis, pues se presenta en el mismo plano de variable probabilística en el que se desarrolla el estimador bayesiano (XPFCEP) que resuelve esta tarea.

En este apartado se analiza primeramente el funcionamiento de la versión básica del “*subtractive*” para descubrir las modificaciones necesarias para conseguir un segmentador que se adapte a las especificaciones de la tesis. El “*subtractive*” modificado de tal forma ha sido denominado “*subtractive enlazado*”, y su funcionalidad se describe también en este apartado.

5.3.1. El “*subtractive*” básico

Debido a que el “*subtractive*” básico es una extensión del “*método mountain*”, el funcionamiento de éste se expone a continuación:

1. Dividir el espacio de definición del conjunto Y a clasificar en una rejilla de $cell_T$ celdas, e inicializar el número de grupos a cero $k = 0$.

2. Calcular la probabilidad de cada celda de la rejilla ($p_{0,cell}$) de ser el centroide de un grupo ($\bar{y}_{cell} \equiv \bar{g}_j$) en función de una medida de distancia ($d_{cell,i}$, normalmente de Manhattan, por ser la más adecuada para el espacio enrejillado) entre la celda \bar{y}_{cell} y cada \bar{y}_i .

3. Elegir la celda con máxima verosimilitud ($p_{\max|iter} = \max(p_{iter,1:cell_T})$) como centroide de una clase j , e incrementar el número de clases ($k = k + 1$). El centroide de la clase j será:

$$\bar{g}_j = \bar{y}_{\max|iter} / p_{\max|iter} = \max(p_{iter,1:cell_T}) \quad <5.23>$$

4. Reducir la verosimilitud de cada celda ($p_{iter,cell}$) en función de su proximidad a la celda elegida como centroide ($d_{cell,\max|iter}$) en el paso 3:

$$p_{iter+1,cell} = p_{iter,cell} - p_{\max|iter} \cdot g(d_{cell,\max|iter}) \quad <5.24>$$

5. Volver al paso 3 mientras el valor máximo de verosimilitud de las celdas de la rejilla esté por encima de un límite ($p_{\max|iter} > p_{th}$).

Si bien el algoritmo descrito es simple y efectivo, conlleva una alta complejidad computacional debida al cálculo de los valores de verosimilitud ($p_{iter,1:cell_T}$), que depende exponencialmente del número de celdas de la rejilla ($cell_T$) y de la dimensión del vector de características usado en el proceso de clasificación λ ($O(\lambda^{cell_T})$).

En 1994 Chiu propone una extensión del “*método mountain*”, a la que llama algoritmo “*subtractive*”. Su funcionamiento es idéntico al expuesto en los párrafos anteriores, excepto por que el espacio de definición del conjunto Y no se enrejilla, y la verosimilitud se calcula para cada punto i de Y , en lugar de para cada celda ($cell$) de la rejilla ($p_{cell} \equiv p_i$). Este cambio conlleva el uso de la distancia euclídea ($d_{2|i,j} = d_{euclidea|i,j}$) o la de Mahalanobis ($d_{Mahalanobis|i,j}$) en lugar de la de Manhattan ($d_{1|i,j} = d_{manhat|i,j}$), para el cálculo de los valores de verosimilitud.

El clasificador “*subtractive*” presenta varias mejoras frente al “*método mountain*”, relacionadas con la diferencia comentada en el párrafo anterior:

- El “*método mountain*” es muy sensible al tamaño de la rejilla elegido para la segmentación. Como este parámetro no existe en el “*subtractive*”, este algoritmo es más robusto.
- Los centroides obtenidos con “*subtractive*” son datos del set ($\bar{g}_{1:k} \subseteq Y$) en lugar de depender del tamaño de la rejilla, por lo que representan de forma exacta a cada grupo.
- El t_{exe} del “*subtractive*” no se multiplica con cada variable incluida en el vector de características para el que se define la segmentación, por lo que, desde este punto de vista este tiempo es menor que el del “*método mountain*”. Además, la complejidad computacional del “*subtractive*” depende únicamente del número de datos m a organizar ($O(m^2)$ según se expone en [Chiu94]), por lo que para sets no demasiado grandes, como el de la aplicación de

MTT, esta suele ser menor que la del “*método mountain*”. Hay que tener en cuenta que la complejidad computacional del “*método mountain*” depende inversamente del tamaño de la rejilla, y éste deberá de definirse relativamente pequeño si se desea suficiente resolución en la clasificación.

El “*método mountain*” fue diseñado para generar una estimación inicial del número de clases k y el valor de sus centroides ($\bar{g}_{1:k}$), con los que inicializar un segmentador como el “*k-medias*”, que requiere esta información como paso previo a su ejecución. En definitiva, el “*método mountain*” fue planteado para resolver el mismo problema que soluciona el proceso de predicción descrito en el apartado anterior como parte del “*k-medias secuencial*” propuesto.

Las características expuestas desaconsejan el uso del “*método mountain*” como clasificador por sí solo. Por el contrario, el “*subtractive*” se encuentra con frecuencia en la literatura científica como segmentador de sets de datos semejantes al de interés ([Chiu96] o [Liu03]).

La funcionalidad de la versión básica del “*subtractive*” se extrae fácilmente a partir de la descripción del “*método mountain*” y de las características que diferencian ambos algoritmos. Los párrafos siguientes exponen esquemáticamente este funcionamiento:

1. Calcular la probabilidad $p_{0,i}$ de cada \bar{y}_i de ser el centroide de una clase ($\bar{y}_i \equiv \bar{g}_j$) en función de una medida de distancia $d_{i,i2}$, entre éste y el resto de elementos de Y ($\bar{y}_{i2} / i2 = 1 : m \not\subset i$). Se obtiene así la PDF característica de Y . La ecuación normalmente utilizada para el cálculo de esta PDF se muestra a continuación :

$$p_{0,i} = \sum_{i2=1}^m e^{-\frac{4}{distA^2} d_{i,i2}^2}, \quad <5.25>$$

donde $distA$ es un factor de ponderación de la distancia $d_{i,i2}$, que tiene por objeto delimitar la zona en el espacio de distancias alrededor de \bar{y}_i , que se considera asociada a éste. De este modo, la verosimilitud de cada dato ($p_{0,i}$) depende en mayor medida de la densidad de puntos en un entorno concreto del mismo. Más adelante se incluye un análisis sobre los efectos que la modificación de $distA$ tiene en el resultado del clustering.

2. Elegir el dato $\bar{y}_{\max|iter}$ de máxima verosimilitud ($p_{\max|iter} = \max(p_{iter,1:m})$) como centroide del grupo j y aumentar el número de grupos ($k = k + 1$). El valor de \bar{g}_j es un punto de Y :

$$\bar{g}_j = \bar{y}_{\max|iter} / p_{\max|iter} = \max(p_{iter,1:m}) \quad <5.26>$$

3. Reducir la verosimilitud de cada \bar{y}_i para la iteración siguiente del proceso (recalcular $p_{iter,i}$), en función de su proximidad al punto elegido como g_j en el paso 2 ($d_{i,\max|iter}$):

$$p_{iter+1,i} = p_{iter,i} - p_{\max|iter} \cdot g(d_{i,\max|iter}) \quad <5.27>$$

La forma de la ecuación normalmente usada para llevar a cabo la resta de la probabilidad es:

$$p_{iter+1,i} = p_{iter,i} - p_{\max|iter} \cdot e^{-\frac{4}{distB^2} d_{i,\max|iter}^2}, \quad <5.28>$$

donde $distB$ es otro factor de ponderación, en este caso de la distancia $d_{i, \max|iter}$, que tiene por objeto delimitar la zona alrededor de \bar{g}_j en la que la verosimilitud de los datos se reduce en mayor medida. El objetivo de este paso es el de evitar que se generen varios centroides, y por tanto, varias clases, en un área del espacio definido de alta densidad respecto al resto. El valor de $distB$ suele establecerse algo mayor que el de $distA$, de modo que los puntos incluidos en el área de asociación definida por $distA$ y en una zona de guarda alrededor de ésta, experimenten una reducción sustancial en su verosimilitud y se establezca una relativa separación entre las clases. Posteriormente se incluye un análisis más profundo sobre la funcionalidad de $distB$. El “*subtractive*” toma su nombre de este paso del proceso, pues traducido al español significa “*restador*”.

4. Volver al paso 2 mientras la verosimilitud $p_{\max|iter} = \max(p_{iter,1:m})$ supere un límite p_{th} :

$$p_{\max|iter} > p_{th}, \quad \langle 5.29 \rangle$$

donde el valor del límite de verosimilitud p_{th} se define de diferentes formas en distintas fuentes bibliográficas ([Chiu94]), que se comentan más adelante en este capítulo. Este paso permite hacer al sistema robusto ante la presencia de outliers, por lo que se incluye también un análisis de su sensibilidad frente a ruidos, en función de su valor.

La Figura 5.31 muestra la funcionalidad del “*subtractive*” básico mediante un diagrama de flujo que resume los pasos expuestos en los párrafos anteriores. La Figura 5.32 muestra un ejemplo de funcionamiento de la versión básica del “*subtractive*” en un espacio bidimensional ($\lambda_T = 2$), empleando la distancia euclídea ($d_{euclídea|i,j}$). Sin necesidad de especificar el número de grupos k en que se desea organizar el set Y , el algoritmo arroja como salida dos grupos ($G_{1:k} \equiv \bar{g}_{1:k} / k = 2$), debido a la densidad de puntos.

Tal y como se aprecia en las imágenes de la Figura 5.32, las clases de salida generadas por el algoritmo se caracterizan, no por sus elementos miembros sino por su centroide ($G_{1:2} \equiv \bar{g}_{1:2}$), que según la funcionalidad de este segmentador, es un dato del set ($\bar{g}_{1:k} \subseteq Y$). Este hecho, propio del carácter no enlazado del clasificador, obliga a introducir una modificación en el proceso que permita definir la asociación concreta de cada dato de Y con alguna de las k clases generadas por el “*subtractive*”, tal y como sí ocurre con el “*k-medias*” básico. A la versión mejorada del “*subtractive*” propuesta en la tesis, se le ha denominado “*subtractive enlazado*” concretamente debido a esta modificación.

En la Figura 5.32 se muestra la evolución secuencial (de la primera iteración en la imagen izquierda a la tercera en la derecha) de la PDF característica del conjunto Y . En azul se presenta la PDF discreta, es decir, la verosimilitud $p_{iter,i}$ de todos los datos ($\bar{y}_i / i = 1:m$) del set Y , en función de su relación de posición en el espacio de segmentación bidimensional definido. En rojo se presenta el dato de máxima verosimilitud en cada iteración del algoritmo ($p_{\max|iter}$), que se convierte en el centroide de un nuevo grupo, en caso de que esta sea mayor al límite p_{th} .

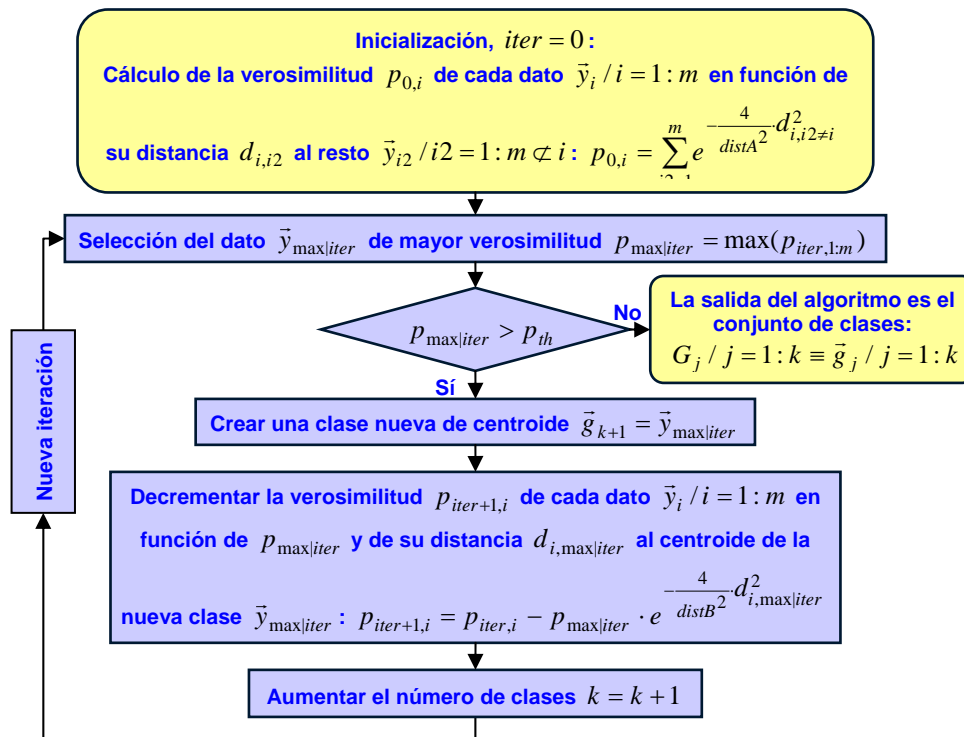


Figura 5.31. Diagrama funcional de la versión básica del “subtractive”.

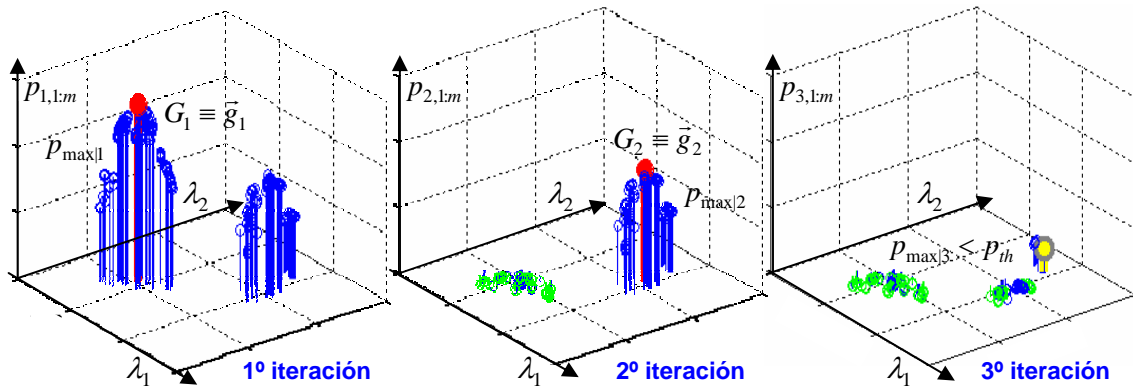


Figura 5.32. Ejemplo de funcionamiento del “subtractive” básico en un espacio bidimensional ($\lambda_T = 2$). Las imágenes muestran la evolución de la PDF que caracteriza al set Y en tres iteraciones. El máximo de la PDF se muestra en rojo si su valor es superior al límite p_{th} .

El efecto que tiene la resta de verosimilitud, incluida como paso 3 en el “subtractive” básico, sobre la PDF inicial se aprecia claramente en la Figura 5.32. Debido a ese proceso y establecido el valor del límite de verosimilitud p_{th} , el punto de máxima verosimilitud en la tercera iteración $p_{max|3}$ no es considerado el centroide de un nuevo grupo (por lo que se marca de forma diferente a los anteriores en la figura) y el segmentador deja de iterar y acaba.

Merece la pena realizar un segundo análisis de funcionamiento del clasificador en el mismo experimento a través de la Figura 5.33, para analizar en detalle su finalización, y el papel de los parámetros $distA$ y $disB$ en ésta. Esta última figura muestra la proyección en el espacio de

distancias ($\lambda_T = 2$) de las imágenes incluidas en la Figura 5.32, en la que se marcan los valores de distancia $distA$ (en línea roja) y $distB$ (en línea azul clara) alrededor de cada uno de los puntos elegidos como centroides $\bar{g}_{1,2}$.

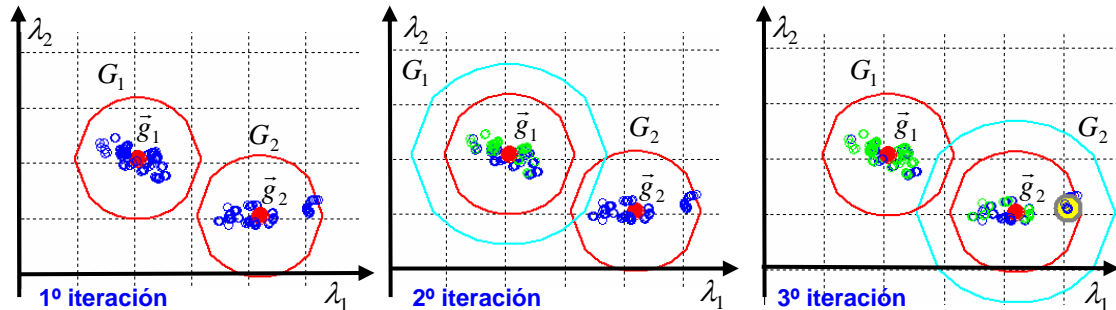


Figura 5.33. Proyección de los resultados mostrados en la Figura 5.32 sobre el espacio bidimensional de clasificación ($\lambda_T = 2$). Con círculo rojo y azul claro se representan, respectivamente, el valor de $distA$ y $distB$ en el espacio de clasificación.

Los círculos rojos encierran a los puntos de Y considerados en mayor medida para obtener la verosimilitud $p_{1,2}$ de $\bar{g}_{1,2}$. Por otro lado, los círculos azules encierran a los puntos que sufren en mayor medida la resta de verosimilitud tras la creación de cada grupo.

En el experimento de la Figura 5.33 se ha fijado $distB = 1.5 \cdot distA$, de modo que se decremente en mayor medida la verosimilitud de los puntos de Y usados para calcular la del centroide y provocar la creación de la clase. De ese modo, tal y como se aprecia en la imagen central de la figura, se elimina la posibilidad de que un conjunto de puntos de gran densidad como el que da lugar al cluster G_1 generen un segundo cluster, y se facilita que lo haga otro conjunto menos denso como G_2 , fuera del radio definido por $distB$ alrededor de \bar{g}_1 .

En la tercera iteración mostrada en la Figura 5.33, el proceso de resta elimina la posibilidad de que el set poco denso a la derecha de \bar{g}_2 (en el eje f_1 del espacio de características) cree un nuevo grupo. Su proximidad a \bar{g}_2 es alta, de modo que el proceso de resta afecta en gran medida a su verosimilitud, convirtiéndola en un valor insuficiente ($p_{\max|3} < p_{th}$) como para producir la creación de un nuevo grupo. Este conjunto de puntos es tratado entonces como parte del más denso (\bar{g}_2).

La relación entre los parámetros $distB$ y $distA$ define, por tanto, una zona en el espacio de características de la clasificación, que tiene, a primera vista, dos funciones:

- *Determinar una zona de separación entre clases*, de modo que se minimicen los errores de duplicación de clases. Es evidente que si la zona de separación se hace muy grande ($distB$ se fija mucho mayor que $distA$), podrán aparecer errores de unión de 2 clases en 1.
- *Determinar una zona de histéresis entre las clases*, de modo que se flexibilice la forma que los datos asociados a las mismas conformen en el espacio de distancias. El proceso de resta

de verosimilitud que incluye el “*subtractive*”, permite cumplir con la especificación de flexibilidad establecida en la tesis, siempre y cuando se asegure la relación $distB > distA$.

A la vista de estos comentarios, en el apartado siguiente se incluye también un análisis de la sensibilidad del segmentador frente al valor de los parámetros $distA$, $distB$ y a su relación.

5.3.2. Análisis de los puntos débiles del “*subtractive*” básico

Una vez expuesto el funcionamiento del “*subtractive*” básico, en este apartado se analizan en detalle los puntos débiles del algoritmo, resueltos en la tesis a través de dos modificaciones:

- La modificación de la versión estándar del segmentador, que da lugar a la propuesta “*subtractive enlazado*”, descrita en el apartado siguiente.
- El algoritmo de validación ya referido en la sección anterior, en la que se presenta el funcionamiento del clasificador “*k-medias secuencial*”.

De nuevo, se usan los tests descritos en el apéndice III, para validar las ideas propuestas.

a) Sensibilidad frente al ruido

Como se ha comentado en párrafos anteriores, en la literatura científica se describen varios métodos de finalización de la versión básica del “*subtractive*”. La diferencia entre uno y otro se encuentra en la definición del parámetro p_{th} . Algunas de las formas más típicas son:

1. p_{th} es una fracción α del primer valor máximo de verosimilitud encontrado en el proceso ($p_{th} = p_{\max|0}$). En este caso y debido al paso de 3 del algoritmo, p_{th} se obtiene del mayor máximo de verosimilitud del proceso global ($p_{th} = \alpha_{new} \cdot p_{\max|0} = \alpha_{new} \cdot p_{\max}$). Utilizando esta forma del parámetro, la ecuación <5.29> se reescribe como se muestra a continuación:

$$p_{\max|iter} > \alpha_{new} \cdot p_{\max} \quad <5.30>$$

2. p_{th} es una fracción α del máximo valor de verosimilitud encontrado en la iteración anterior. En este caso la ecuación <5.29> se reescribe como sigue:

$$p_{\max|iter} > \alpha_{new} \cdot p_{\max|iter-1} \quad <5.31>$$

3. p_{th} es un valor constante. Este caso es totalmente equivalente al primero si p_{th} es además adaptativo, por lo que no requiere un análisis específico.

En cualquiera de los dos primeros casos descritos, el parámetro α_{new} tiene una relación directa con la robustez del “*subtractive*” frente a los outliers presentes en el set Y . Debido a ello parece interesante hacer un análisis de su sensibilidad frente al ruido, en función de α_{new} .

La gráfica de la Figura 5.34 muestra la tasa de error del “*subtractive*” básico en función del método de cálculo elegido para el límite de creación de una nueva clase p_{th} . Los resultados se

han obtenido ejecutando los dos algoritmos en el experimento complejo. La opción 2 de cálculo del error del límite de verosimilitud (en verde en la figura) produce un error de duplicado de clusters, que no aparece apenas en la opción 1 (en rojo), y que se debe al recálculo del límite p_{th} en cada iteración del algoritmo. De hecho el segundo método puede generar problemas de convergencia, que no aparecen si se elige el primero, aunque el segmentador sea recursivo.

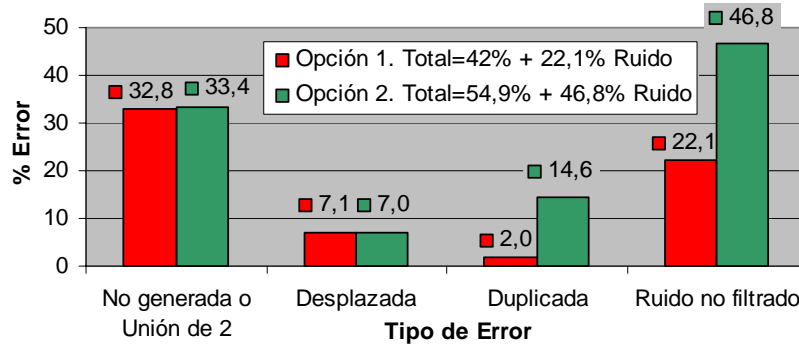


Figura 5.34. Tasa de error del “subtractive” básico en función del método usado para el cálculo del límite de verosimilitud p_{th} . Los resultados de sensibilidad al ruido se obtienen con test de ruido y el resto con el complejo.

Por otro lado, en la misma Figura 5.34 se incluye la tasa de error en el filtrado de outliers obtenido al ejecutar el “subtractive” básico con las dos opciones de cálculo de p_{th} comentadas en el experimento de ruido. Como se aprecia, el método 1 tiene más del doble de efectividad en el filtrado de ruido que el método 2. La opción 1 parece más acertada para el cálculo de p_{th} .

En cualquiera de los casos, es evidente que cuanto menor es el valor del parámetro α menos restrictivo es el algoritmo a la hora de crear clases asociadas a sets de baja densidad, ya sean outliers o grupos de baja verosimilitud. El objetivo es buscar la alternativa que sea menos sensible a modificaciones del parámetro α_{new} , de modo que el ajuste de este parámetro sea lo menos dependiente posible del experimento.

La Figura 5.35 muestra la sensibilidad del parámetro en lo referente a la tasa de error por duplicados y clases no generadas, extraída del testeo de los dos métodos en el experimento complejo. No se incluyen errores de unión de 2 clases en 1 ni de desplazamiento en la gráfica, pues ambas opciones presentan tasas semejantes de estos tipos de error.

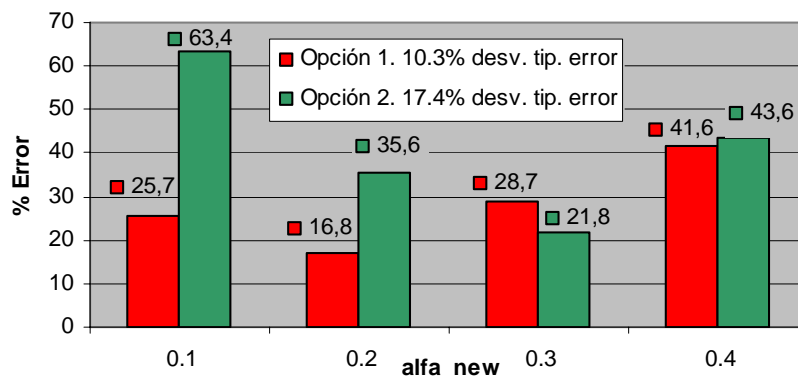


Figura 5.35. Tasa de error total del “subtractive” básico en el experimento complejo, en función del método de cálculo del límite p_{th} , para distintos valores del parámetro α_{new} (alfa_new).

b) Sensibilidad frente a variaciones de $distA$ y $distB$

La sensibilidad de la versión básica del “*subtractive*” al valor de los parámetros $distA$ y $distB$ se desprende fácilmente de la descripción de su funcionalidad, incluida en párrafos anteriores. Este problema de sensibilidad es análogo al que presenta el algoritmo “*k-medias extendido*” frente al parámetro $distM$, pues la funcionalidad de los parámetros es semejante: el número de grupos k_i generados por el segmentador, así como el tamaño del espacio al que se asocia cada una de ellas, dependen de $distA$ y $distB$. Se incluye aquí, por tanto, un análisis de esta sensibilidad, por las mismas razones por las que se hizo con el algoritmo booleano.

La Figura 5.36 muestra la tasa de error del “*subtractive*” básico en el experimento complejo para distintos valores de $distA$ (con una variación del 40% del parámetro). Esta figura es muy semejante a la Figura 5.9, que describe la sensibilidad del “*k-medias secuencial*”. Los errores obtenidos con la ejecución del segmentador se han agrupado en dos tipos en la Figura 5.36, para observar fácilmente como según se incrementa el valor del parámetro disminuye el número de errores por clases desplazadas o duplicadas, pero aumenta el de clases no generadas.

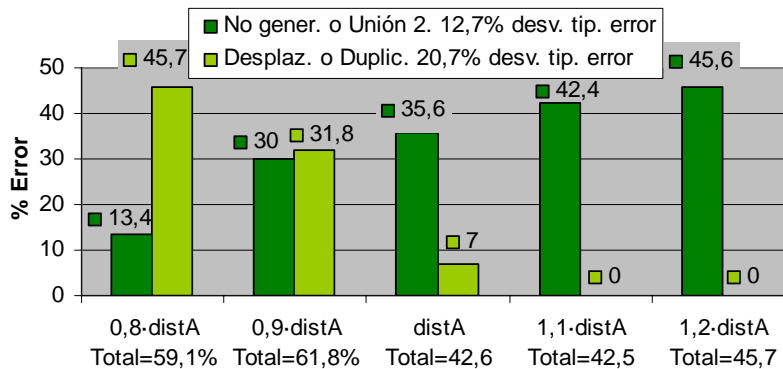


Figura 5.36. Tasa de error del “*subtractive*” básico en el experimento complejo, en función del valor de $distA$.

La Figura 5.36 demuestra que el problema de sensibilidad del “*subtractive*” frente al valor de $distA$ es semejante al que presenta el “*k-medias secuencial*” frente al valor de $distM$, si bien la desviación típica del error mostrado en la figura es menor a la que presenta el “*k-medias secuencial*” en la Figura 5.9. Este hecho se debe al funcionamiento borroso del “*subtractive*”, es más robusto a la modificación de sus parámetros, como se comenta en párrafos anteriores.

Hay que añadir que, aunque no se aprecia en la Figura 5.36, los errores generados por “*subtractive*” ante un valor bajo de $distA$ son sobre todo de desplazamiento, menos importantes que los de duplicación generados por “*k-medias secuencial*” en la misma situación. Este hecho se debe a que la resta de verosimilitud incluida en el flujograma del “*subtractive*” elimina la mayor parte de errores de duplicación de clases, pero no todos.

Por otro lado, será necesario introducir alguna mejora que permita disminuir algo más la sensibilidad del clasificador respecto de $distA$. El algoritmo de validación, propuesto como solución al problema de sensibilidad del “*k-medias secuencial*”, mejora también la sensibilidad del “*subtractive*”, haciendo que su fiabilidad dependa en menor medida de un perfecto ajuste del valor de sus parámetros ($distA$ y también $distB$ y α_{new}), como se demuestra más adelante.

La Figura 5.37 muestra la tasa de error arrojada por el algoritmo en el mismo experimento, ante variaciones de $distB$. Para obtenerla se ha fijado $distA$ a su valor de ajuste más adecuado y se ha modificado $distB$ del 100% al 200% del valor de $distA$, ya que la descripción teórica su indica que $distB \geq distA$. De esta figura se extraen las siguientes conclusiones:

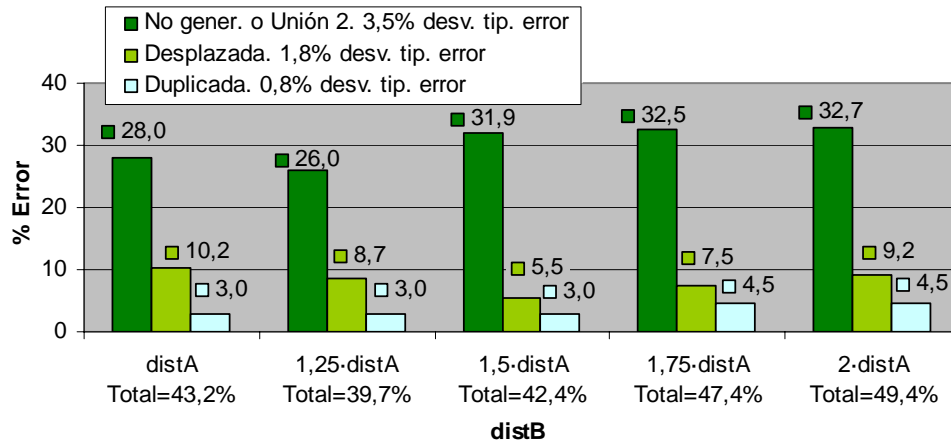


Figura 5.37. Tasa de error del “subtractive” básico en el test complejo, en función de $distB$, ajustado $distA$.

- El “subtractive” básico tiene una sensibilidad mínima respecto al parámetro $distB$, una vez fijado $distA$, y manteniendo siempre $distB > distA$.
- Un aumento del valor del parámetro produce un incremento del número de clases no generadas, y en contrapartida, un decremento del número de clases desplazadas. Este hecho está relacionado con el significado teórico del parámetro, expuesto en párrafos anteriores: $distB$ determina indirectamente la separación mínima que debe existir entre clases.
- La tasa de error debida a clases duplicadas permanece prácticamente constante. Esto significa que la modificación de la zona de histéresis, determinada por la relación entre $distA$ y $distB$, no permite flexibilizar del todo el comportamiento del segmentador al tipo y forma de las clases, en el espacio de clasificación. La especificación de flexibilidad se mejora en la propuesta de la tesis, mediante un algoritmo de anulación de verosimilitud.

Con todo ello se concluye que, como se comenta en puntos anteriores, la propia funcionalidad del “subtractive”, minimiza su sensibilidad respecto de sus parámetros, si se compara con la de algoritmos de diferente base teórica como el “ k -medias”. Esta conclusión se extrae del análisis de la Figura 5.9, donde se observa que la desviación típica de la tasa de error generada por el “ k -medias extendido” ante una variación del 40% del parámetro como la aquí analizada, es del 13.5% en el caso de la duplicación de clases, y del 23.9% en el de la unión de 2 clases en 1; valores mayores a los que presenta la Figura 5.36, que muestra la tasa del error “subtractive” ante su parámetro menos estable: $distA$.

c) Asignación de miembros

Como se comenta en párrafos anteriores, el “subtractive” básico, al contrario que el “ k -medias”, es un segmentador no enlazado. Esto significa que la información que proporciona el

clasificador sobre cada grupo $G_{j,t} / j=1:k_t$ no incluye una lista con los datos asignados al mismo L_j . Esta información es usada por el XPFPCP propuesto en la tesis, por lo que hay que modificar el “*subtractive*” básico y hacerlo enlazado, obteniéndose el “*subtractive enlazado*”.

En el apartado siguiente se describe el método utilizado para hacer que el “*subtractive enlazado*” propuesto sea un proceso enlazado.

5.3.3. Versión del algoritmo desarrollada en esta tesis: el “*subtractive enlazado*”

A la vista de los análisis expuestos en el apartado anterior, se propone implementar una nueva versión del “*subtractive*” que cumpla las especificaciones de la aplicación de MTT. Como ya se ha comentado, la nueva versión del algoritmo se ha denominado “*subtractive enlazado*”, y las aportaciones que incluye son las siguientes:

1. Se incluye un proceso de asignación de miembros a cada clase, generándose de este modo una versión enlazada del “*subtractive*”. Esta mejora da nombre al segmentador propuesto.
2. Se aumenta la flexibilidad del algoritmo con un proceso de anulación de verosimilitud de los datos próximos al \bar{g}_j de cada clase $G_j / j=1:k$, en el espacio de interés, que se añade al paso de resta de verosimilitud del “*subtractive*” básico.
3. Se elige como límite de verosimilitud $p_{th} = \alpha_{new} \cdot p_{max}$, para hacer más robusto el sistema frente a outliers, tal y como prueban los análisis expuestos en el apartado anterior.

En los apartados siguientes se describe la función de los dos primeros, aún no expuesta.

a) Proceso de asignación de miembros

Dentro de los distintos métodos que se pueden utilizar para elegir el conjunto de miembros de cada grupo $L_j / j=1:k_t$ a partir del resultado proporcionado por el “*subtractive*” básico ($G_{1:k} \equiv \bar{g}_{1:k}$), el más directo consiste en utilizar sus parámetros de distancia $distA$ y $distB$.

Normalmente, el proceso consistirá en asignar al grupo j los puntos de Y que se encuentren a una distancia de \bar{g}_j menor a la establecida por estos parámetros, formando el subconjunto L_j . De este modo, el enlazado de los datos de Y a los grupos se hace utilizando el mismo espacio de características y de distancias usado para hacer la clasificación.

A partir de esta definición se pueden utilizar varias combinaciones de los dos parámetros de distancia ($distA$ y $distB$) para concretar el proceso de asignación de miembros. Sin embargo, dada la funcionalidad de $distA$ en el “*subtractive*”, parece lógico utilizar este parámetro en el proceso de asignación, con lo que quedaría definido por la expresión siguiente:

$$\bar{y}_i \in L_j / d_{i,j} < distA \tag{5.32}$$

Conviene comentar dos cuestiones relativas al método de selección de los miembros:

- Teniendo en cuenta la relación funcional y cuantitativa existente entre los parámetros $distA$ y $distB$, podría parecer lógico establecer distintos valores de pertenencia de los datos de Y , en función de su ubicación en el espacio de segmentación. Para ello, se podría recurrir a un método semejante al que se propuso para borrosificar la asignación de los datos en el “ k -medias”, basado en el funcionamiento del algoritmo TTSAS (ver el apartado 5.2.2 y la Figura 5.7). Sin embargo, esto no tendría sentido en este caso, pues que la verosimilitud p_i asignada a cada dato de Y , contiene ya esta información de forma más precisa.
- Por otro lado, en el proceso finalmente propuesto no se utiliza la verosimilitud de cada dato p_i para determinar su valor de pertenencia a cada grupo, pues en la aplicación de interés no se requiere una información de pertenencia precisa.

El hecho de no asignar los datos de Y a cada uno de los clusters con un valor de pertenencia concreto no conlleva la eliminación del carácter borroso de la clasificación resultante, es decir, que cada \bar{y}_i pueda asignarse a ninguno, uno o más clusters.

La Figura 5.38 muestra el resultado obtenido al ejecutar el “*subtractive*” básico con el algoritmo de asignación propuesto sobre dos conjuntos de datos un espacio de segmentación bidimensional ($\lambda_r = 2$). En ella se demuestra que se mantiene el carácter borroso del clustering.

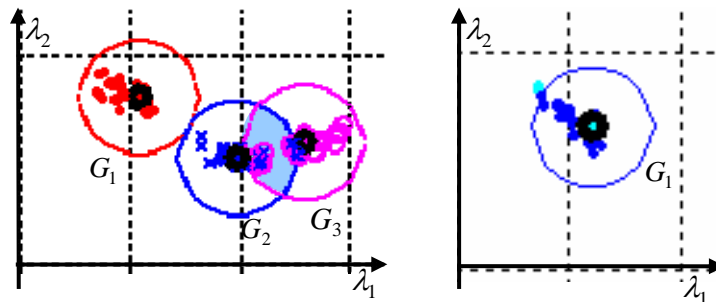


Figura 5.38. Ejemplos de funcionamiento del proceso de enlazado propuesto para el “*subtractive*” básico sobre dos conjuntos de datos. La pertenencia de cada \bar{y}_i a una u otra clase, y la representación del parámetro $distA$, se muestra con colores. La zona de pertenencia borrosa de la imagen izquierda entre G_2 y G_3 aparece en azul claro.

La pertenencia de cada punto a ninguna (azul claro), uno o más de un grupo se muestra en función del color y la forma de representar cada \bar{y}_i (puntos rojos, cruces azules o círculos malvas). Los círculos en azul oscuro representan la definición del método de asignación propuesto en función del parámetro $distA$ para cada grupo.

En la imagen derecha de la Figura 5.38 aparecen algunos puntos que no se han asociado a ninguna clase en azul claro. En la imagen izquierda, se ha pintado en el mismo color la zona de intersección de pertenencia que aparece entre las clases G_2 y G_3 al utilizar el parámetro $distA$ para establecer la asociación de datos a cada grupo. Los puntos de Y localizados en esa zona en el espacio de segmentación son asociados a los dos grupos, según el método de asignación expuesto. La segmentación generada es, por tanto, borrosa.

En otros trabajos ([Kaufman90] o [Everitt01]), se describe un proceso diferente al aquí expuesto para establecer la asignación de los datos a organizar a los grupos. El método consiste en asignar a cada grupo aquellos puntos que sufren una disminución de su verosimilitud en el paso 3 del “*subtractive*” básico superior a cierto límite.

Al margen de que este método implica la definición de un nuevo parámetro, incluyendo por tanto, un nuevo factor de sensibilidad en la funcionalidad del clasificador, los resultados obtenidos son algo peores a los que se consiguen con el método propuesto, en lo que a valores de separación y distorsión de los grupos se refiere, tal y como se observa en la Figura 5.39.

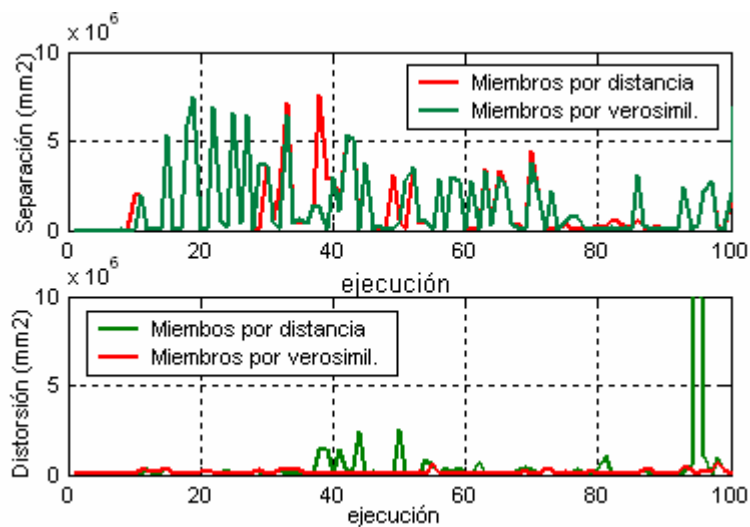


Figura 5.39. Gráficas comparativas de los valores de separación y distorsión que arroja el “*subtractive*” en el experimento complejo, con el método de asignación de miembros propuesto (basado en la distancia, en rojo), y con el propuesto en [Everitt01] (basado en la reducción de la verosimilitud, en verde).

La separación de los grupos es ligeramente mayor con el método propuesto en algunas situaciones, y la distorsión de las mismas es notablemente menor, sobre todo en situaciones en las que el conjunto Y está muy concentrado, y su pertenencia es más difícil de definir.

Por todo ello, se propone utilizar el método de asignación inicialmente presentado en el “*subtractive enlazado*” propuesto en esta tesis. Al añadir este proceso al “*subtractive*” básico, la salida del clasificador será un conjunto k de grupos, caracterizados de la siguiente forma:

$$G_j / j = 1:k \equiv \{\vec{g}_j, L_j / j = 1:k\} \quad <5.33>$$

b) Proceso de anulación de verosimilitud

El objetivo ya mencionado del proceso de anulación, es el de minimizar la sensibilidad del “*subtractive*” al tamaño, forma y desigual densidad de las clases en el espacio de segmentación.

El proceso de anulación consiste en cancelar en la PDF del set Y , en la que se basa el “*subtractive*”, la verosimilitud p_i de los datos que se suponen relacionados con cada grupo j creada, cada vez que se crea un nuevo grupo: $p_i = 0 / \vec{y}_i \rightarrow G_j$.

A partir de la definición de funcionalidad anterior se podría analizar cuáles han de ser los datos a tener en cuenta en el proceso de anulación, del mismo modo que se hizo en el apartado anterior. Sin embargo, una vez definido el proceso de asignación de miembros al cluster parece evidente que los datos del conjunto Y , cuya verosimilitud ha de ser cancelada son los miembros del cluster, es decir:

$$p_i = 0 / \bar{y}_i \in L_j \tag{5.34}$$

La ecuación anterior implica que los elementos del set Y asociados a un grupo no pueden convertirse en centroides de otro nuevo, lo cual no significa que no puedan ser miembros de otro próximo en el espacio usado. Es decir, el proceso de anulación de verosimilitud así descrito no afecta a la borrosidad de la clasificación resultante.

El algoritmo expuesto tampoco modifica la zona de histéresis alrededor de cada clase definida por la relación entre $distA$ y $distB$, y por tanto no conlleva un aumento de la tasa de error por unión de 2 clases en 1, como se demuestra en el mas adelante en el documento. Sin embargo, sí elimina casi totalmente los errores de duplicación, que se generan en la segmentación cuando las clases a discriminar en el conjunto Y son de tipología diversa.

Esta mejora disminuye además, el t_{exe} del clasificador (ver apartados posteriores), pues gracias a ella se puede evitar el recálculo de la densidad de los puntos que la tienen anulada, $p_{iter+1,i} = 0 / i = 1 : m$.

En la Figura 5.40 se muestra la evolución de la PDF creada por el “*subtractive*” básico con (imágenes superiores) y sin (imágenes inferiores) proceso de anulación, en una escena real.

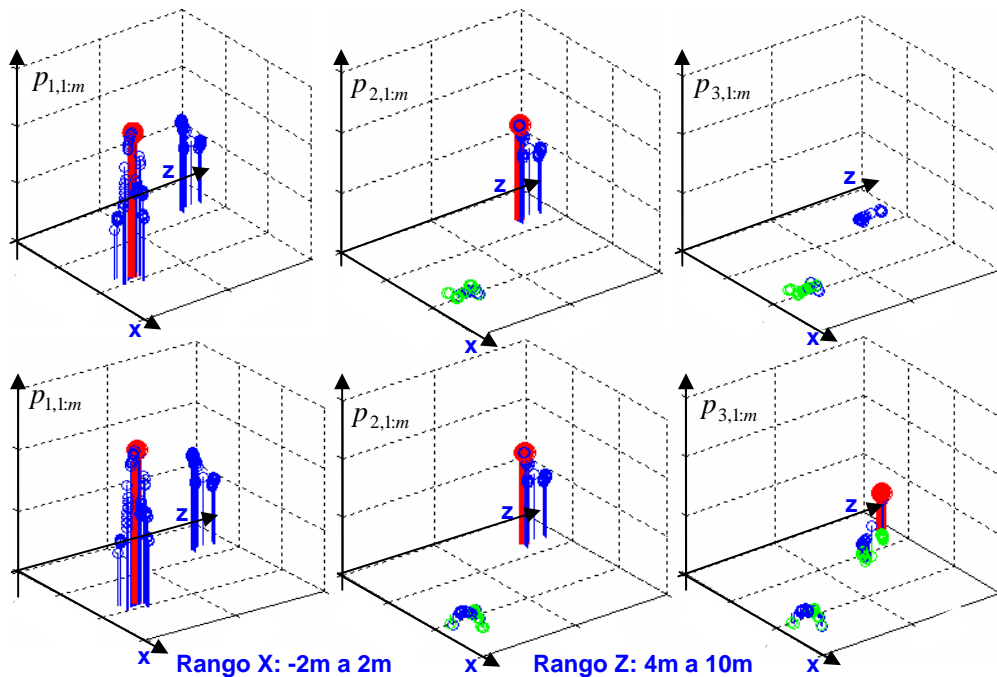


Figura 5.40. Resultados arrojados por el “*subtractive*” básico con anulación de verosimilitud (imágenes superiores) y sin él (imágenes inferiores), en una escena real. En cada fila se muestra la evolución en tres iteraciones de la PDF normalizada que caracteriza al set de medidas Y . Su máximo se muestra en rojo si es superior al límite p_{th} .

En la escena mostrada en la Figura 5.40 aparecen dos grupos de puntos con distribución muy diferente, uno asociado a una papelera, y otro asociado a una persona (ver la Figura 5.41⁷). Tal y como se aprecia en la figura, el clasificador sin proceso de anulación duplica el grupo asociado a la papelera, mientras que el que cuenta con proceso de anulación no lo duplica.

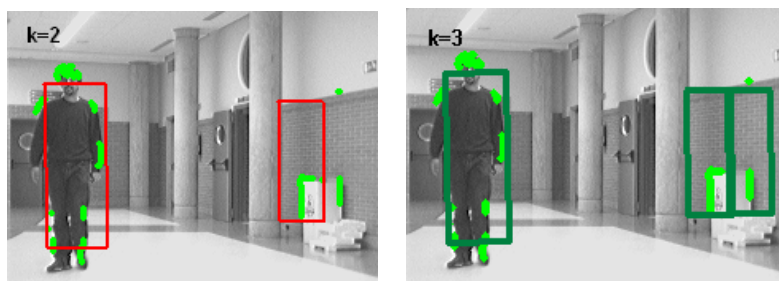


Figura 5.41. Proyección sobre el plano imagen de los puntos de Y (en verde claro), y de los resultados de clasificación mostrados en la Figura 5.40. La imagen izquierda muestra el resultado correcto, generado por el “subtractive” con anulación de verosimilitud, y la de la derecha el generado por el “subtractive” básico, que presenta un error de duplicación de una clase.

Estas situaciones de desigualdad en la distribución de la densidad set Y son típicas en la aplicación de interés, como se comenta en apartados anteriores. El aumento de flexibilidad del “subtractive”, conseguido con el proceso de anulación es, por ello, especialmente interesante.

Finalmente, la Figura 5.42 muestra el diagrama de flujo del “subtractive enlazado” propuesto. En ella se han marcado con fondo verde las aportaciones introducidas al “subtractive” básico, que se analizan en los párrafos anteriores.

5.3.4. Proceso de validación de clases

Para incrementar su robustez frente al ruido y su sensibilidad frente a la modificación de los parámetros, se añade al “subtractive enlazado” un proceso de validación de clases.

Este algoritmo es semejante al que se describe de forma general y aplicado al “ k -medias secuencial” en el apartado 5.2.5. Su implementación implica la gestión de la variable de validación $\beta_{j,t}$ para cada clase $G_{j,t} / j=1:k_t$ en función de: su verosimilitud $p_{j,t}$, y su ubicación en el espacio de segmentación respecto a una posición predicha de ésta, a partir de la ubicación anterior de todas las clases ($d_{t,j,jold} / jold=1:k_{t-1}$).

Como se desprende del párrafo anterior y se comenta ya en el apartado 5.2.5, el proceso de validación se basa en el análisis de la evolución temporal de cada clase $G_{j,t} / j=1:k_t$ a través de su parámetro de identidad τ_j . Esta identificación temporal, que facilita la aplicación de la validación al “ k -medias secuencial” al ser implícita en ese caso, no está presente en el flujograma del “subtractive enlazado” definido hasta este punto (Figura 5.42).

⁷ Los dibujos muestran la proyección de los resultados sobre la imagen obtenida con el sistema de sensado usado. Como en la Figura 5.27, el círculo que representa a cada clase generada por los clasificadores en XZ , y que queda como un cilindro al prolongarse en la coordenada de altura (y), se proyecta como un rectángulo en la imagen.

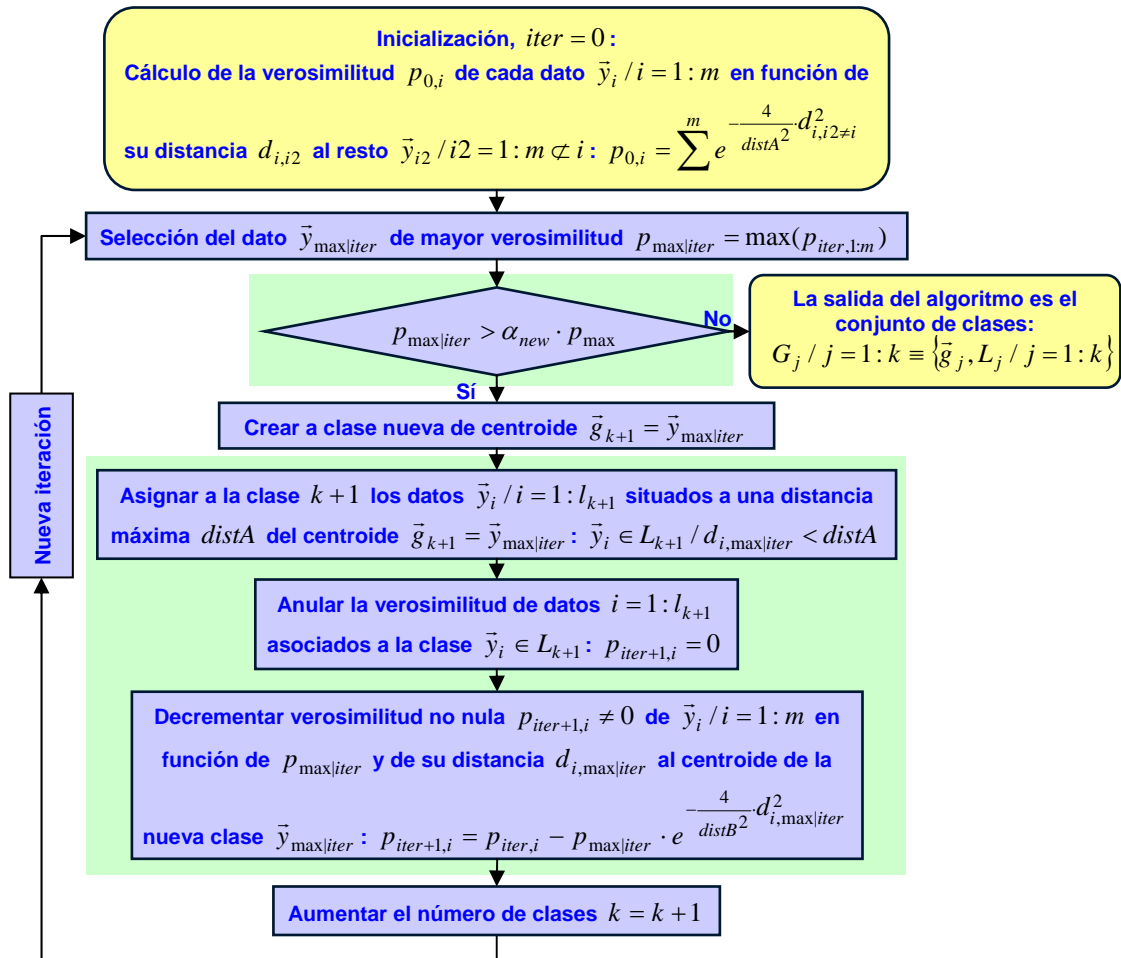


Figura 5.42. Diagrama funcional del “subtractive enlazado”. Las mejoras aportadas al flujoograma del “subtractive” básico se destacan con fondo verde.

La incorporación del proceso de validación a este último pasa, por tanto, por la implementación en el clasificador borroso de una etapa de identificación de cada clase $G_{j,t} / j = 1 : k_t$ a través de una variable τ_j idéntica a la generada por el “*k-medias secuencial*”. Al añadir a éste la validación, la información referente a cada clase que el “*subtractive enlazado*” genera a su salida se completa respecto a lo mostrado en <5.33>, hasta coincidir con la dada por el “*k-medias secuencial*” (ver expresión <5.20>).

En este punto se describe el funcionamiento específico de los procesos de identificación y validación que completan el flujoograma del “*subtractive enlazado*” con validación. También se incluye un análisis funcional de estas mejoras.

a) El proceso de identificación de clases del “*subtractive enlazado*”

Para desarrollar el algoritmo de identificación comentado se emplea el mismo propuesto para el “*k-medias secuencial*”, en el apartado 5.2.4. Resumiendo lo allí expuesto, este método se basa en la propagación de las clases $G_{1:k,t}$ en el tiempo a través de su $\bar{g}_{1:k,t}$, y usando una modificación del modelo dinámico CV (el \widehat{CV}) que caracteriza al movimiento de los targets en la aplicación de MTT. El proceso de identificación referido incluye las siguientes partes:

1. Una etapa de cálculo del valor predicho de los centroides $\bar{g}_{0,1:k,t} = \bar{g}_{1:k,t|t-1}$ de las clases, caracterizada por la expresión <5.16> que se repite aquí por claridad:

$$\bar{g}_{0,j,t} = \bar{g}_{j,t|t-1} = f_j(\bar{g}_{j,t-1}, \bar{u}_{j,t-1}) / j = 1:k_t, \quad <5.35>$$

donde $f_j()$ es la expresión general de la función de transición del modelo \widehat{CV} aplicado a la clase $j = 1:k_t$, del que $\bar{u}_{j,t}$ es el vector de entrada (<5.12> es la expresión específica).

2. La etapa de identificación propiamente dicha, que al contrario de lo que sucede con el “*k-medias secuencial*”, no está implícita en la funcionalidad del “*subtractive enlazado*”. Con objeto de emular el proceso de identificación del segmentador booleano, se propone identificar cada clase $G_{j,t} / j = 1:k_t$ con aquella $G_{jold,t-1} / jold = 1:k_{t-1}$ cuyo centroide predicho $\bar{g}_{jold,t|t-1}$ esté en el espacio de clasificación más próximo al de la primera $\bar{g}_{j,t}$, y viceversa. Para medir la proximidad entre los centroides se emplea la distancia euclídea $d_{euclídea|j,jold}$, pues es la medida de similitud usada en la segmentación. La distancia resultante se denomina $d_{t,j,jold}$, y su descripción matemática viene dada por la expresión:

$$d_{t,j,jold} = \min(d_{euclídea|j,1:k_{t-1}}) / j = 1:k_t \quad <5.36>$$

Cada clase $G_{j,t} / j = 1:k_t$ creada en la ejecución en t del “*subtractive enlazado*” toma el identificador de alguna de las $G_{jold,t-1} / jold = 1:k_{t-1}$ existentes en la ejecución en $t-1$ del segmentador $\tau_j = \tau_{jold}$; o un identificador nuevo $\tau_j = \tau_{new}$ en caso de que G_j no pueda ser relacionada con ninguna de las $G_{jold,t-1}$. Este último caso se produce cuando la distancia $d_{t,j,jold}$ supera un límite $distI$, cuyo significado y valor se comenta más adelante. El proceso ha de mantener la unicidad del identificador de cada clase, por lo que la asignación entre cada cluster $G_{j,t}$ y alguno del conjunto $G_{jold,t-1}$ ha de ser unívoca.

3. Una etapa de obtención o actualización de $\bar{u}_{j,t}$ del modelo \widehat{CV} aplicado a cada clase $G_{j,t} / j = 1:k_t$, dada por la expresión <5.17> que se repite aquí por claridad:

$$\bar{u}_{j,t} = g_j(\bar{g}_{j,t}, \bar{g}_{j,t-1}) / j = 1:k_t, \quad <5.37>$$

donde $g_j()$ es una función genérica (ver en <5.15> una descripción más específica) que representa, para cada clase $j = 1:k_t$, la relación entre $\bar{u}_{j,t}$ y la evolución de la posición de \bar{g}_j actualizada por el segmentador, una vez identificada la relación entre $G_{j,t}$ y $G_{jold,t-1}$, es decir, entre $\bar{g}_{j,t}$ y $\bar{g}_{jold,t-1}$

El sistema así obtenido es el “*subtractive enlazado*” con identificación. Su flujograma se muestra en la Figura 5.43 y tiene una funcionalidad idéntica a la del “*k-medias secuencial*” de la Figura 5.13, pues genera para cada clase la misma información (ver expresión <5.18>):

$$G_{j,t} / j = 1 : k_t \equiv \{\bar{g}_{j,t}, L_{j,t}, \tau_j / j = 1 : k_t\} \quad \langle 5.38 \rangle$$

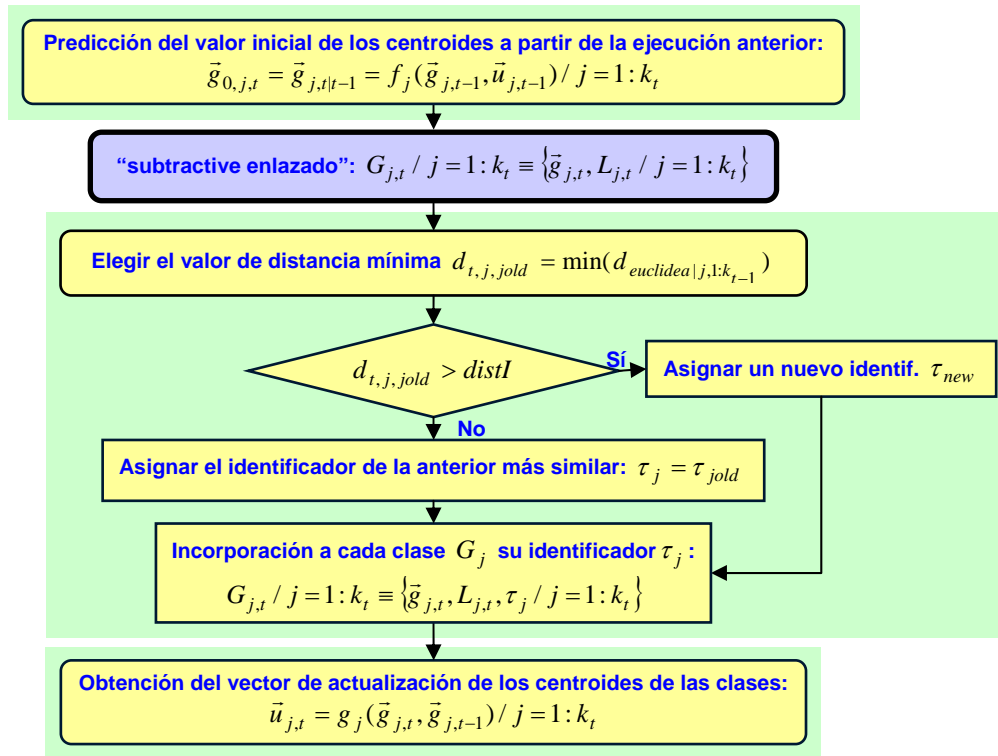


Figura 5.43. Diagrama funcional del “subtractive enlazado” con proceso de identificación (en amarillo). Las diferentes etapas que lo conforman se diferencian con fondo verde. El bloque azul sustituye al flujograma del “subtractive enlazado” tal y como aparece en la Figura 5.42.

El proceso de identificación se realiza, por tanto, usando la característica de distancia $d_{t,j,jold}$ empleada en la etapa de validación del clasificador. De esta forma, se facilita la implementación de la validación en el “*subtractive enlazado*”, al igual que ocurre en el “*k-medias secuencial*”, y se minimiza el incremento de carga computacional del segmentador que ésta supone, tal y como se demuestra en el apartado siguiente.

Además, la identificación así definida deja clases existentes en $t-1$ sin asociación en t , y crea en t clases con identificadores no asignados en $t-1$. El proceso de validación permite finalmente eliminar o validar estas clases de forma robusta, completando así la tarea de identificación perseguida.

Por otro lado, y siguiendo el razonamiento expuesto en el punto 5.2.4 anterior, el empleo de un proceso de identificación para el “*subtractive enlazado*” semejante al desarrollado para el “*k-medias secuencial*” permite usar el clasificador borroso como algoritmo de asociación de tipo NN a usar en la tarea de MTT de interés.

El hecho de que la identificación implementada para el “*subtractive enlazado*” sea explícita, en lugar de implícita como ocurre en el “*k-medias secuencial*”, disminuye la fiabilidad de esta tarea, y por ende del proceso de asociación que éste desarrolla y es imprescindible en el funcionamiento del seguidor multimodal, objetivo de la tesis. Como consecuencia de esta falta

de robustez pueden producirse intercambios de identidad en las clases, y si estos se aplican a la asociación, errores en la evolución del estimador fatales en el tracking. En el punto siguiente se muestra un ejemplo de este tipo de errores.

El razonamiento expuesto en el párrafo anterior desaconseja el uso del clasificador borroso frente al booleano como algoritmo de asociación en la tarea de MTT de interés. Sin embargo, el efecto de robustecimiento que el proceso de validación tiene en la identificación permite validar la fiabilidad de ésta, tal y como demuestra el análisis de los resultados de identificación generados por el “*subtractive enlazado*”, expuesto más adelante.

b) El proceso de validación de clases aplicado al “*subtractive enlazado*”

A partir de la información generada por el proceso de identificación diseñado para el “*subtractive enlazado*”, la incorporación al segmentador de la etapa de validación definida, en su forma genérica, en el apartado 5.2.5 es fácil de implementar. Por ello, al igual que sucede con el “*k-medias secuencial*”, esta incorporación no produce un incremento en el t_{exe} del segmentador global, tal y como se demuestra en el apartado siguiente, lo que facilita la consecución de la especificación de tiempo real perseguida en la tesis.

El cálculo de las dos variables que, según lo expuesto en el punto 5.2.5, permiten actualizar el parámetro de validación $\beta_{j,t}$ de cada clase $G_{j,t} / j=1:k_t$ se realiza en este caso como sigue:

1. La característica de proximidad $d_{t,j,jold}$ a usar en la validación no ha de ser calculada, pues se obtiene en el proceso de identificación, tal y como se describe en el apartado anterior, a través de la expresión <5.36>.
2. Por su parte, para asignar un valor de verosimilitud $p_{j,1,t}$ a cada grupo $G_{j,t} / j=1:k_t$ parece lógico usar los valores de la PDF que representa el set Y_t , base de funcionamiento del clasificador borroso. Una primera opción es asignar a cada clase un valor de verosimilitud instantánea $p_{j,t}$, igual a la de su miembro centroide $p_{\max|iter,t}$, a partir del que ésta se crea (ver la Figura 5.42). Aplicándole a éste $p_{j,t} = p_{\max|iter,t}$ el factor de olvido correspondiente α_{valid} (ver la ecuación <5.19>) se obtiene a través de la expresión <5.39> el parámetro de validación buscado $p_{j,1,t}$. Por otro lado, si se considera el carácter enlazado de la versión propuesta en la tesis, parece lógico usar como valor de verosimilitud instantánea $p_{j,t}$ de cada clase, la suma de densidades iniciales de los puntos asociados a ésta: $p_{j,t} = \sum_{i=1}^l p_{0,i,t}$. Aplicando este valor a la expresión genérica de filtrado por α_{valid} , se obtiene una segunda opción de cálculo de $p_{j,1,t}$, dada por la ecuación <5.40>:

$$p_{j,1,t} = \alpha_{valid} \cdot p_{\max|iter,t} + (1 - \alpha_{valid}) \cdot p_{j,1,t-1} / j=1:k_t \quad <5.39>$$

$$p_{j,1,t} = \alpha_{valid} \cdot \sum_{i=1}^{i=l} p_{0,i,t} + (1 - \alpha_{valid}) \cdot p_{j,1,t-1} / j=1:k_t \quad <5.40>$$

Con objeto de decidir cuál de las dos opciones de cálculo del parámetro de validación $p_{j,t}$ es mejor, se pone a prueba el “*subtractive enlazado*” con un proceso de validación definido por $d_{t,j,jold}$ (dado por la expresión <5.36>) y por cada uno de los dos valores propuestos para $p_{j,t}$. Los resultados de fiabilidad generados en el experimento complejo, por las dos versiones del algoritmo así definidas se muestran en la Figura 5.44.

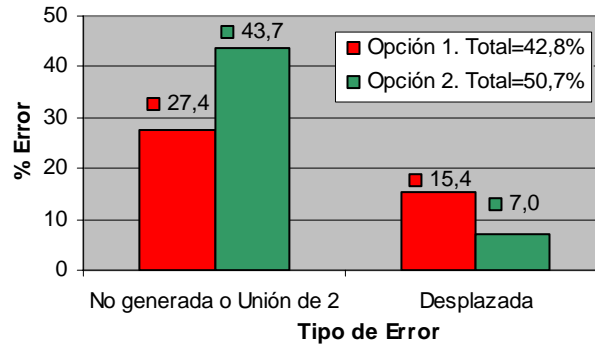


Figura 5.44. Tasa de error del “*subtractive enlazado*” en función del método usado para la obtención del valor de verosimilitud instantánea de la clase $p_{j,t}$, en el experimento complejo.

Tal y como se extrae de la Figura 5.44, la extrapolación de la verosimilitud del centroide a la instantánea de la clase, $p_{j,t} = p_{\max|iter,t}$ (aplicación de la ecuación <5.39>), minimiza el error de no generación de clases y unión de 2, si bien produce un mayor número de clases desplazadas que la aplicación de la ecuación <5.40>, en la que $p_{j,t} = \sum_{i=1}^{l_j} p_{0,i,t}$. Teniendo en cuenta la mayor relevancia de primero de los errores frente al segundo, se opta por usar la expresión <5.39> para calcular el parámetro de verosimilitud de la clase.

Además, analizando el resultado en detalle, se observa que la diferencia de tasa de error se produce con distribuciones desiguales de datos en el espacio de clasificación, muy frecuentes en la aplicación de interés. En esos casos, el uso de la opción 2 en el cálculo de la verosimilitud de la clase favorece la validación de las clases formadas por una alta densidad de puntos, invalidando las de pocos puntos. La opción 2 de cálculo de $p_{j,t}$ no es adecuada para esta tesis.

El proceso de validación a añadir al “*subtractive enlazado*” queda así totalmente definido. La Figura 5.45 muestra la especificación del flujograma del proceso de validación genérico (Figura 5.16) al ser aplicado al “*subtractive enlazado*”, en el que se marcan con fondo verde los pasos que cambian respecto al genérico.

Además, en la Figura 5.46 se muestra el flujograma del “*subtractive enlazado*” con procesos de identificación y validación, obtenido al añadir al diagrama de la Figura 5.43 (simplificado) una llamada al de la Figura 5.45. La salida de este algoritmo, para cada clase $G_{j,t}$, es idéntica a la obtenida del “*k-medias secuencial*” con validación (ver ecuación <5.20>) y es de gran utilidad en la aplicación del segmentador al proceso de estimación multimodal, como se explica en capítulos posteriores:

$$G_{j,t} / j = 1:k_t \equiv \{\bar{g}_{j,t}, L_{j,t}, \tau_j, \beta_{j,t}, p_{j,t} / j = 1:k_t\} \quad <5.41>$$

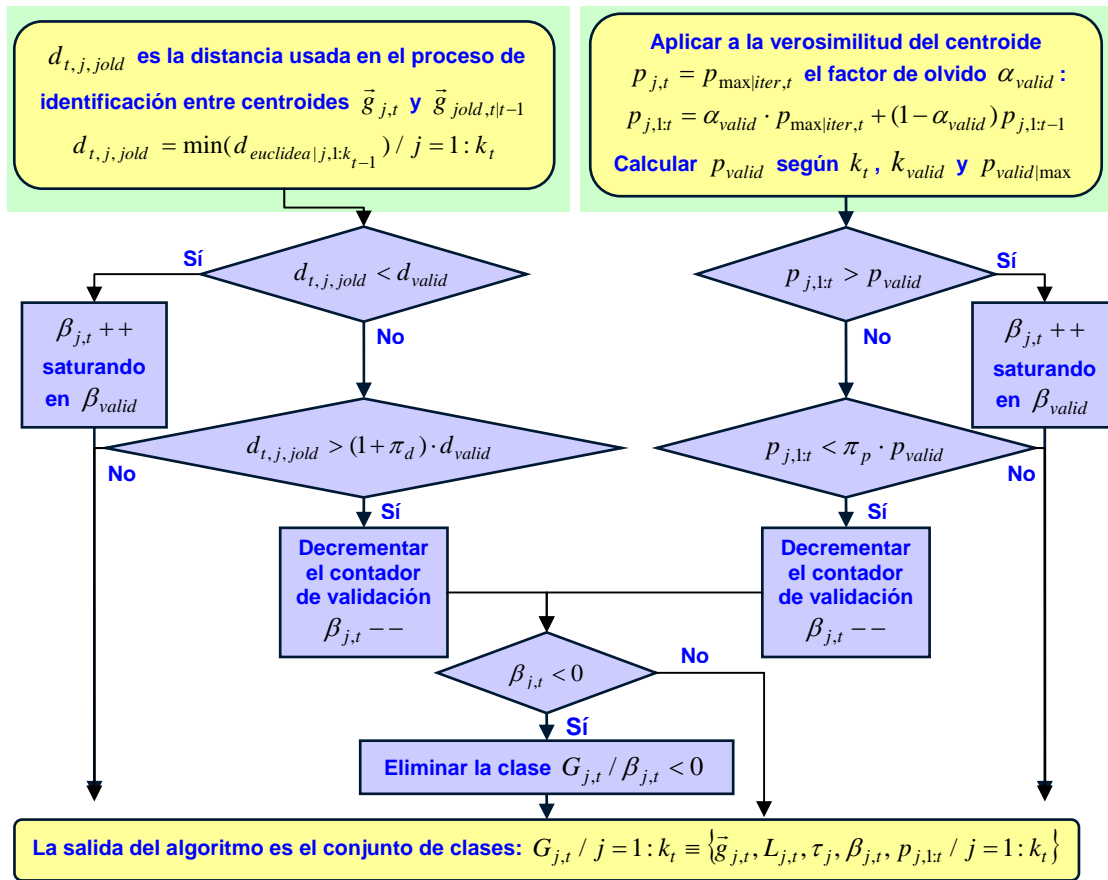


Figura 5.45. Diagrama funcional del proceso de validación genérico aplicado al “subtractive enlazado” con proceso de identificación. Las etapas modificadas respecto al flujograma de la versión genérica (Figura 5.16) se muestran con fondo verde.

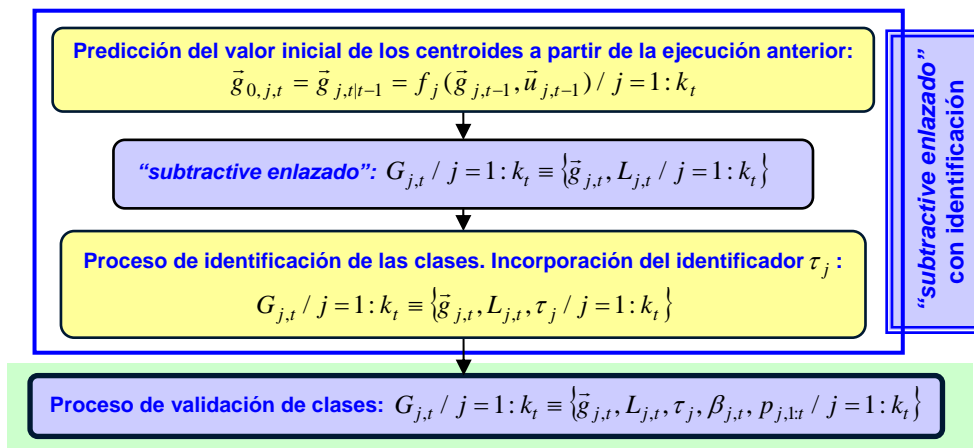


Figura 5.46. Diagrama funcional del “subtractive enlazado” con validación. La figura se obtiene al añadir al flujograma del “subtractive enlazado” con proceso de identificación (Figura 5.43 simplificada) el proceso de validación de la Figura 5.45 (sobre fondo verde).

c) Análisis funcional del proceso de validación aplicado al “*subtractive enlazado*”

En este punto se incluye un análisis de los efectos más importantes del proceso de validación aplicado al “*subtractive enlazado*”, completando el incluido en el apartado 5.2.5, donde se ejemplifica su funcionalidad general y en su aplicación “*k-medias secuencial*”.

Como se comenta a lo largo de este capítulo, el proceso de validación mejora la fiabilidad del clasificador y lo robustece frente al ruido y frente a variaciones de sus parámetros. Tal y como se extrae del análisis incluido en el apartado 5.2.5, cada una de las dos condiciones de validación tiene una implicación específica en estas mejoras. Así, la condición de verosimilitud de la validación (fijada por p_{valid} , π_p y α_{valid}) aplicada al “*subtractive enlazado*” tiene un efecto de robustecimiento del clasificador frente a cambios en la densidad del set Y y frente al ruido idéntico al que presenta el algoritmo al ser añadido al “*k-medias secuencial*”, mostrado en el apartado 5.2.5 (ver la Figura 5.21).

Por otro lado, en el mismo ejemplo de aplicación de la validación al “*k-medias secuencial*” del apartado 5.2.5, se demuestra también el efecto de robustecimiento que la condición de distancia tiene en el proceso de identificación implícito en ese clasificador (Figura 5.20). En su aplicación al “*subtractive enlazado*”, esta condición tiene un efecto semejante, si bien al ser diferente la funcionalidad del algoritmo de identificación añadido explícitamente al clasificador borroso, las mejoras en la robustez del proceso son menores.

Con objeto de que la validación por distancia aporte al proceso de identificación del “*subtractive enlazado*” el mismo robustecimiento a la par que flexibilidad, frente a modificaciones de la dinámica de movimiento de las clases que aporta al del “*k-medias secuencial*”, se endurece la condición que la produce, fijada por d_{valid} y π_d , y se relaja la de identificación fijada por $distI$, haciendo $distI = (1 + \pi_d) \cdot d_{valid}$. De esta forma, la validación a través de la condición de distancia se produce si se cumple:

$$d_{t,j,jold} < d_{valid} \Rightarrow d_{t,j,jold} < \frac{distI}{(1 + \pi_d)}, \quad \langle 5.42 \rangle$$

expresión obtenida a partir de la definición establecida para $distI$, y de la condición de validación por distancia, $d_{t,j,jold} < d_{valid}$, expuesta en la Figura 5.45.

Por su parte, las clases $G_{jold} / jold = 1:k_{t-1}$ generadas en $t-1$ que no han sido identificadas en ninguna de las creadas en t son invalidadas por incumplimiento de la condición de identificación ($d_{t,j,jold} < distI$ de la Figura 5.43), que coincide con la de distancia ($d_{t,j,jold} > (1 + \pi_d) \cdot d_{valid}$ de la Figura 5.45) al fijar el valor de $distI = (1 + \pi_d) \cdot d_{valid}$:

$$d_{t,j,jold} > (1 + \pi_d) \cdot d_{valid} \Rightarrow d_{t,j,jold} > distI \quad \langle 5.43 \rangle$$

Teniendo en cuenta que π_d es un parámetro de valor normalizado, es evidente que la condición de identificación es menos restrictiva que la de validación según muestra la ecuación

<5.42>, y que las clases que incumplen la de identificación incumplen también la de validación, según se observa en la expresión <5.43>.

Este análisis demuestra que la validación por distancia tiene un efecto de robustecimiento del proceso de identificación del “*subtractive enlazado*”. En el apartado siguiente se expone un ejemplo de este efecto, y de otros ya comentados que el proceso de validación completo tiene en la funcionalidad del clasificador finalmente obtenido.

A la hora de fijar el límite de validación por probabilidad p_{valid} en el “*subtractive enlazado*”, a partir de los parámetros $p_{valid|max}$ y k_{valid} , es necesario atender a ciertas consideraciones. Tal y como se expone en el punto anterior, el valor de verosimilitud de la clase $p_{j,t}$ se obtiene del de su centroide $p_{j,t} = p_{max|iter,t}$ (a través de la expresión <5.39>). Además, este valor $p_{max|iter,t}$ es a su vez el que ha dado lugar a la creación de una clase al superar el umbral p_{th} dependiente del parámetro α_{new} ($p_{max|iter,t} > p_{th} = \alpha_{new} \cdot p_{max,t}$, ver apartado 5.3.2). Es evidente, que la condición de validación por probabilidad ha de ser menos restrictiva que la de creación de clases, y por tanto el valor de p_{valid} ha de ser menor al de p_{th} .

5.3.5. Resultados comparativos del “*subtractive enlazado*” con proceso de validación respecto al “*subtractive*” básico

Del mismo modo que se hizo en el apartado 5.2.6 con el algoritmo “*k-medias secuencial*” propuesto, en este apartado se comparan los resultados que arroja la versión propuesta del algoritmo “*subtractive*”, el “*subtractive enlazado*”, con la versión básica del mismo, en los experimentos descritos en el apéndice III, para obtener conclusiones respecto a las mejoras de robustez, fiabilidad y tiempo de cómputo que aporta la versión propuesta.

a) Tasa de error:

La Figura 5.47 muestra la tasa de error obtenida en la ejecución del experimento complejo (a) y del completo corto (b), del “*subtractive enlazado*” sin proceso de identificación ni de validación (en tonos rojos) y del “*subtractive*” básico (en tonos verdes). A la vista de ésta, se extraen las siguientes conclusiones:

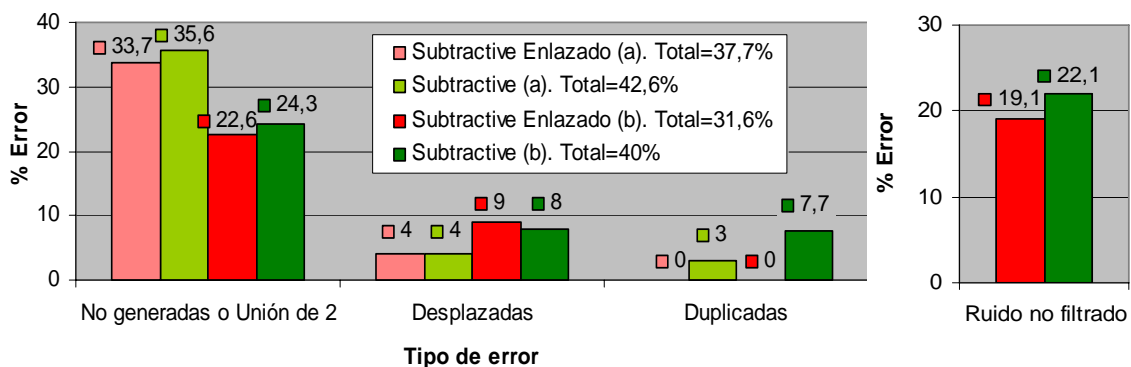


Figura 5.47. Tasa de error de los algoritmos “*subtractive enlazado*” (en tonos rojos) y “*subtractive*” básico (en tonos verdes) en los experimentos complejo (a) y completo corto (b).

- La tasa de error total del “*subtractive enlazado*” es algo menor a la del “*subtractive*” básico en los dos experimentos probados (37.7% frente a 42.6% en el experimento complejo, y 31.6% frente a 40% en el experimento completo corto). El algoritmo propuesto presenta, por tanto, una mayor fiabilidad que el básico aún sin validación.
- Un análisis más profundo revela que el segmentador propuesto elimina los errores de duplicación de clusters producidos por el “*subtractive*” básico gracias al proceso de anulación de verosimilitud, tal y como se expone en el apartado 5.3.3 anterior. Este hecho es sobre todo evidente en situaciones como la mostrada en la anterior Figura 5.41, en las que se producen cambios bruscos y momentáneos de la densidad de datos relacionados con un cluster. La flexibilidad del “*subtractive enlazado*” para hacer frente a estas situaciones, y por ende, a la diversa tipología de los objetos implicados en la tarea de seguimiento de interés en la tesis queda, por tanto, probada. El proceso de validación, sin embargo, mejora aún más el comportamiento del clasificador, como se comenta en el punto anterior.
- En contrapartida a lo anterior, el proceso de anulación de la verosimilitud produce un ligero desplazamiento de los $\bar{g}_{1:k,t}$ en el espacio de clasificación, tal y como se aprecia también en la Figura 5.41. Sin embargo, este efecto es mínimo (el segmentador propuesto solo incrementa en un 1% la tasa de clusters desplazados en el experimento completo largo, respecto a la obtenida con la versión básica del mismo).
- Finalmente, la tasa de error por la unión de 2 en 1, generada por el “*subtractive enlazado*” es menor que la generada por la versión básica del algoritmo (alrededor de un 2.2% en ambos experimentos). Este resultado demuestra que el proceso de anulación de verosimilitud no solo no conlleva el aumento de esta tasa de error, como se adelanta en el apartado 5.3.3, sino que, apoyado por la correcta selección del umbral de verosimilitud para la generación de clases p_{th} , mejora el comportamiento del clasificador en este aspecto, fundamental, por otra parte en la aplicación de MTT de interés.
- En lo que se refiere a la tasa de rechazo al ruido, el “*subtractive enlazado*” tiene también un mejor comportamiento que el “*subtractive*” básico; el primero filtra el 81.9% de los ruidos, mientras que el segundo lo hace en el 77.9% de las ocasiones, en el experimento de ruido. Esta mejora se produce gracias a la correcta elección de la fórmula de cálculo del límite de verosimilitud p_{th} , tal y como se muestra en el punto 5.3.2 anterior. En cualquier caso, esta mejora se amplía de nuevo al añadir la validación al clasificador.

Con la información expuesta, es posible concluir que el “*subtractive enlazado*” mejora la fiabilidad del “*subtractive*” y su capacidad de filtrar outliers.

Por otra parte, en la Figura 5.48 se muestra la misma gráfica de tasas de error de la Figura 5.47, obtenida en este caso al comparar los resultados generados por el “*subtractive enlazado*” sin (en colores rojos) y con proceso de identificación y de validación (en colores amarillos), a la vista de la cual se extraen nuevas conclusiones:

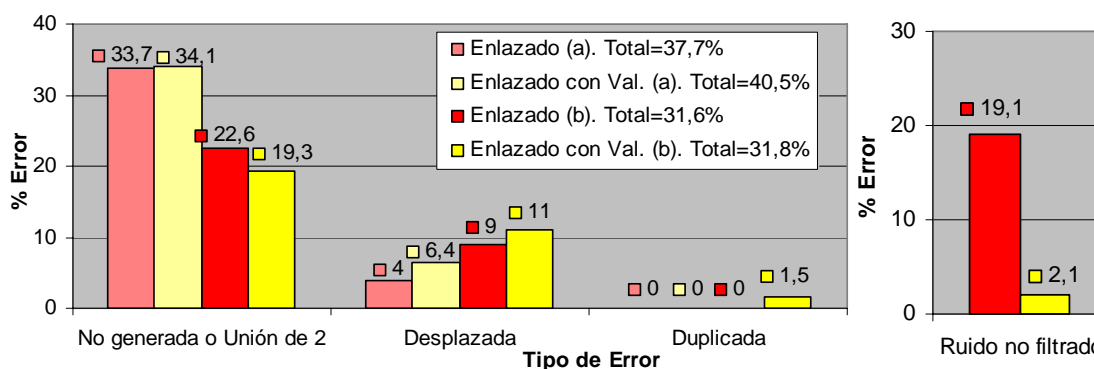


Figura 5.48. Tasa de error del algoritmo “subtractive enlazado” sin (en tonos rojos) y con proceso de identificación y de validación (en tonos amarillos) en los experimentos complejo (a) y completo corto (b).

- En este caso, la tasa total de error del clasificador con validación no disminuye, sino que aumenta ligeramente en el experimento complejo (un 2.8%, del 37.7% al 40.5%), y queda más o menos al mismo valor (pasa del 31.6% al 31.8%) en el completo corto. Esta conclusión es semejante a la extraída en el apartado 5.2.6 al comparar la fiabilidad del “*k-medias secuencial*” con y sin validación ante los mismos conjuntos de prueba (Figura 5.23). En aquel caso, la validación disminuye la tasa de clases duplicadas, que en este caso se mantiene prácticamente nula. En este caso, la tasa de clases desplazadas, en lugar de disminuir, aumenta con la validación, haciendo que la tasa total también aumente.
- La razón del aumento de clases desplazadas (que también se produce en una pequeña cantidad en el caso del clasificador booleano, como muestra la Figura 5.23) se encuentra en el efecto que el proceso de validación tiene sobre las clases de densidad baja y cambiante, especialmente importante para este clasificador que basa su funcionamiento en esta variable. En estos casos, el “*subtractive enlazado*” sin validación elimina las clases, mientras que el que incluye validación las mantiene pero las desplaza, tal y como se muestra más adelante en este punto. Efectivamente, la tasa de clases no generadas o unión de 2 en 1, más importante que la de desplazamiento, se decrementa en el experimento completo corto en un 3.3%, a la vez que ésta última aumenta en un 2%.
- Sin embargo el efecto más importante del proceso de validación en la fiabilidad del “*subtractive enlazado*” aparece, al igual que ocurre con el “*k-medias secuencial*”, en el filtrado del ruido. La tasa de rechazo al ruido del segmentador con validación alcanza el 97.9%, frente al 81.9% del algoritmo sin validación.

A la vista de los resultados obtenidos, la conclusión global del análisis de fiabilidad es, de nuevo, semejante a la extraída en referencia al “*k-medias secuencial*” en el apartado 5.2.6: el proceso de validación tiene un efecto muy importante en el filtrado de outliers (eliminándolos en casi el 98% de los casos), cumpliendo así con el objetivo fundamental para el que se diseña en lo que a la fiabilidad del segmentador se refiere; sin embargo, al margen del rechazo al ruido, son las mejoras aportadas a la versión básica del clasificador las que permiten disminuir su tasa de error de manera importante. En el caso analizado en este apartado, la versión enlazada del “*subtractive*” mejora en alrededor de un 7% la fiabilidad del segmentador en su versión básica.

Al margen del filtrado de outliers, otro objetivo fundamental de la incorporación de la validación al “*subtractive enlazado*” es su insensibilización frente a los parámetros característicos del algoritmo, cuestión que es analizada en los párrafos siguientes.

En la Figura 5.49 se muestra la tasa de error obtenida al repetir el experimento analizado en la anterior Figura 5.36, con objeto de estudiar la sensibilidad, en este caso, del “*subtractive enlazado*” con proceso de validación⁸ ante el parámetro *distA*. A la vista de la figura es fácil concluir que el segmentador es más robusto frente a modificaciones de este parámetro, pues las tasas de error generadas por el algoritmo se estabilizan en determinados rangos de modificación del parámetro, modo que se flexibiliza la elección del valor de *distA*, y por ende, el comportamiento del clasificador ante el tamaño de las clases en el espacio de clasificación.

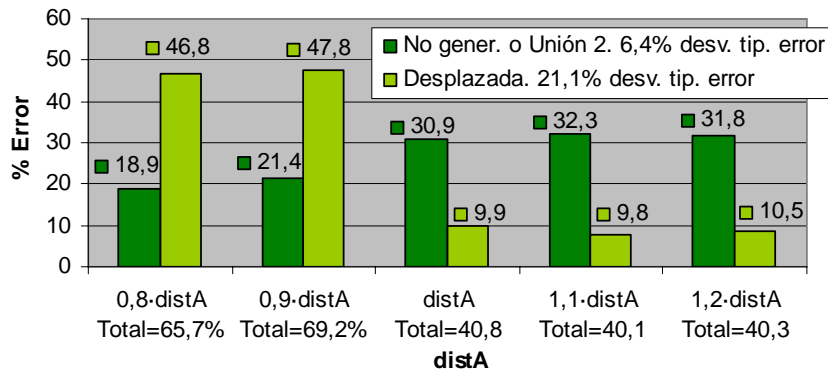


Figura 5.49. Tasa de error del “*subtractive enlazado*” con procesos de identificación y validación en el experimento complejo, en función del valor de *distA*.

La validación produce, por tanto, el efecto de robustecimiento que se espera, y que también genera a ser incorporada al flujograma del “*k-medias secuencial*”.

Por su parte, la prueba de sensibilidad del “*subtractive enlazado*” con identificación y validación frente al parámetro *distB* no se incluye aquí, ya que arroja valores de desviación típica de la tasa de error frente a variaciones del parámetro igual de pequeños a los generados por el algoritmo sin validación (mostrados en la anterior Figura 5.37). Esto permite concluir que el “*subtractive enlazado*” es en cualquier caso bastante insensible al valor de *distB*.

La Figura 5.50 permite, a su vez, analizar la sensibilidad del “*subtractive enlazado*” con identificación y validación frente a modificaciones del parámetro α_{new} ⁹, usado para establecer el umbral de verosimilitud de creación de una clase p_{th} (ver expresión <5.30>). La figura es equivalente a la Figura 5.35, en la que se muestran los resultados generados en la misma prueba por el “*subtractive*” básico, con éste y otro método de cálculo del umbral p_{th} .

Tal y como se observa en la Figura 5.50, la desviación típica del error generado por el clasificador con validación para los distintos valores de α_{new} no se modifica en gran medida respecto a la obtenida con el mismo algoritmo sin validación. En este caso se obtiene una

⁸ No se incluye la tasa de error debida a la duplicación de clases pues es nula para cualquier valor del parámetro.

⁹ No se incluye la tasa correspondiente a errores de desplazamiento y unión de 2 clases en 1, pues alcanza un valor de alrededor del 12% independientemente del valor asignado a α_{new} .

desviación típica media del 10.5%, mientras que de los resultados mostrados en la Figura 5.35 en referencia a esta opción de cálculo de p_{th} se extrae un valor del 10.3% de desviación.

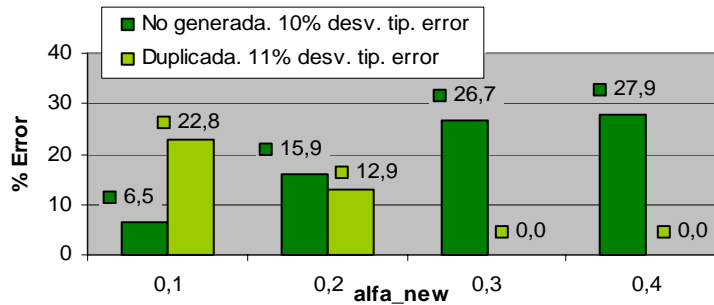


Figura 5.50. Tasa de error del “subtractive enlazado” con procesos de identificación y validación en el experimento complejo, en función del valor del parámetro α_{new} (alfa_new).

En cualquier caso, las tasas de error en la Figura 5.50 se estabilizan para valores de α_{new} mayores a 0.2, por lo que puede concluirse que el algoritmo no es sensible a las modificaciones del valor del parámetro dentro de ese rango. Además merece la pena destacar que para ese rango de valores de α_{new} , la tasa de error por duplicación de clases se anula, demostrándose así el efecto que tiene este método de cálculo del umbral p_{th} comentado en el apartado 5.3.2.

Finalmente, la Figura 5.51 muestra la misma gráfica de desviación típica de la tasa de error generada por el “subtractive enlazado” con procesos de identificación y validación en el experimento complejo, ante variaciones de 0 a 1 de los cuatro parámetros normalizados implicados en la validación: α_{valid} , $p_{valid|max}$, π_p , π_d . De nuevo esta gráfica es equivalente a la Figura 5.25 en la que se analiza la sensibilidad, en ese caso, del “k-medias secuencial” con validación frente a la misma situación.

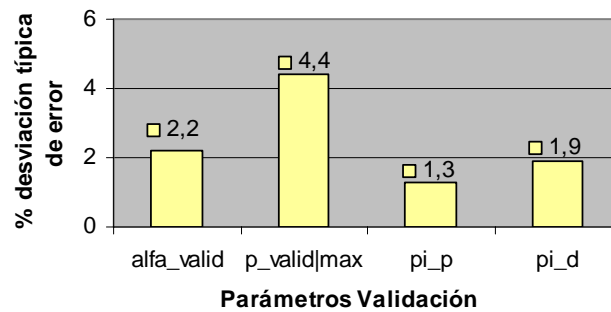


Figura 5.51. Desviación respecto a la media de error del “subtractive enlazado” con identificación y validación en el test complejo para los parámetros: α_{valid} (alfa_valid), $p_{valid|max}$ (p_valid|max), π_p (pi_p) y π_d (pi_d).

La Figura 5.51 muestra que la sensibilidad del clasificador borroso frente al valor de estos parámetros (α_{valid} , $p_{valid|max}$, π_p , π_d) es pequeña, presentando los resultados obtenidos una desviación sobre la media de error del 2.5%.

En lo que a los parámetros no normalizados se refiere, únicamente se analiza la sensibilidad del clasificador frente al parámetro β_{valid} , con un experimento idéntico al usado para obtener la

Figura 5.51¹⁰. En ese caso se obtiene un valor de desviación respecto a la media de error de solo el 1.9%, modificando el valor del parámetro de 1 a 5, lo que demuestra la robustez del “*subtractive enlazado*” con validación también frente a este parámetro.

Las conclusiones expuestas confirman que el proceso de validación robustece el comportamiento del “*subtractive enlazado*”, y demuestran que se cumple el segundo de los objetivos para los que se diseña el proceso de validación.

Para completar el análisis de fiabilidad se estudia la tasa de error generada por el proceso de identificación diseñado para el “*subtractive enlazado*” con y sin el efecto de la validación. Este análisis se hace del mismo modo que el equivalente realizado sobre el “*k-medias secuencial*” en el apartado 5.2.6. El resultado se muestra en la Figura 5.52.

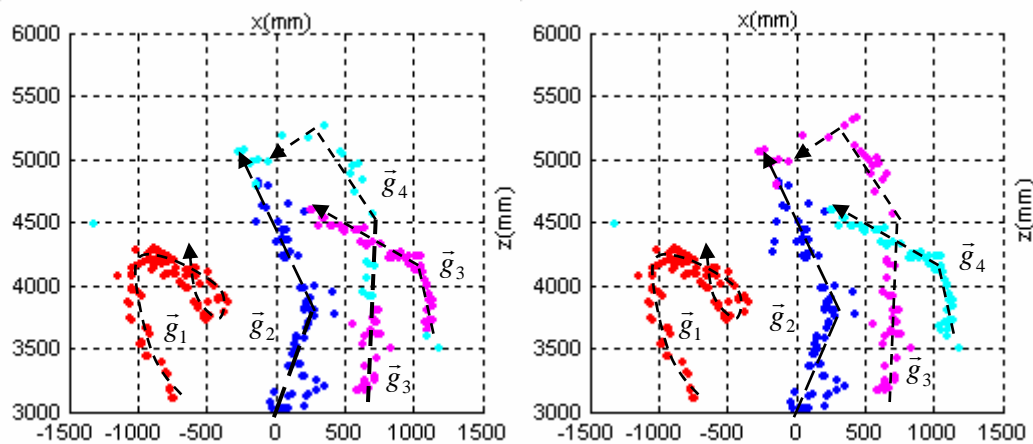


Figura 5.52. Representación por colores de la posición de $\bar{g}_{1:4}$, proyectada en el plano XZ , de cada objeto identificados por el “*subtractive enlazado*” con identificación y sin validación (imagen izquierda) / con validación (imagen derecha). Los resultados se obtienen ejecutando los algoritmos en la segunda parte del test complejo.

La Figura 5.52 muestra los resultados de identificación obtenidos ejecutando el “*subtractive enlazado*” con proceso de identificación pero sin (imagen izquierda) y con validación (imagen derecha) en la segunda parte del experimento complejo. En la figura se representa por colores, en función de su identidad, la posición en el plano XZ de $\bar{g}_{1:4}$ asociados a cada uno de los objetos en la escena a lo largo del tiempo. Sobre la secuencia de puntos coloreados se muestra el “*ground truth*” manual usado para comparar los resultados de identificación mediante una flecha que indica además la dirección del movimiento de los objetos.

De los resultados expuestos en la Figura 5.52 se concluye que la validación mejora en gran medida la identificación realizada por el proceso descrito en la Figura 5.43: el clasificador sin validación intercambia la identidad de G_3 y G_4 en el 50% de ejecuciones del test, debido a un problema de oclusión (el objeto asociado a, G_3 cuyo centroide se marca en color magenta en la Figura 5.52, ocluye al asociado a G_4 , de centroide representado en azul claro, al pasar antes por

¹⁰ El análisis de sensibilidad no se realiza para los otros dos parámetros de validación no normalizados (d_{valid} y k_{valid}) pues, como se comenta en el apartado anterior, tienen un valor de ajuste teórico.

el punto de cruce que éste último). La tasa de error de identificación arrojada por el “*subtractive enlazado*” con identificación y sin validación alcanza, por ello, el 10.4%, frente a la generada por el clasificador con validación que es del 0.5%, y se produce sobre G_3 al final de la prueba.

La validación robustece, por tanto, el funcionamiento del proceso de identificación. Esta conclusión es similar a la que se extrae de la Figura 5.26 en el apartado 5.2.6, si bien en este caso el efecto de la validación es más importante que en aquel, pues al no estar su funcionamiento implícito en el del clasificador, el proceso de identificación tiene menor fiabilidad si se usa en el “*subtractive enlazado*” que si se hace en el “*k-medias secuencial*”.

Por otro lado, en las dos imágenes de la Figura 5.52 se observa que la posición de $\bar{g}_{2,3}$ presenta a lo largo de todo el test un desplazamiento importante respecto al “*ground truth*” manual marcado. La generación de este tipo de error es propia de todas las versiones del clasificador borroso, como demuestran los resultados expuestos anteriormente.

Además, en la Figura 5.52 se confirma la correcta funcionalidad del proceso de predicción de centroides base de la identificación incluso sin validación. Del mismo modo que sucede en su aplicación al “*k-medias secuencial*” (apartado 5.2.6), este proceso es flexible a cambios bruscos de dinámica como el que sufre el objeto asociado a G_1 , a pesar de la sencillez del modelo de predicción en la que se fundamenta.

Al margen del problema de desplazamiento, que como ya se analiza más adelante, tiene una relevancia menor en la aplicación de interés, el “*subtractive enlazado*” con identificación y validación realiza la tarea de identificación cumpliendo con las especificaciones planteadas.

b) Tiempo de ejecución y número de iteraciones:

Como se comenta en apartados anteriores, el t_{exe} del “*subtractive*” básico es elevado si se compara con el del otro clasificador analizado, el “*k-medias*” básico. Este hecho supone que el uso del “*subtractive*” en la aplicación de interés dificulta la consecución de la especificación de tiempo real establecida.

Por otro lado, cualquier mejora del clasificador diseñada con el objetivo de reducir su t_{exe} conlleva renunciar a su funcionamiento básico: la complejidad computacional de la versión básica del algoritmo $O(m^2)$, se debe fundamentalmente al cálculo de la densidad de cada \bar{y}_i a clasificar $p_{iter,i} / i = 1:m$ (ver apartado 5.3.1), y esta variable es imprescindible en la funcionalidad del algoritmo, como se extrae del análisis expuesto en los puntos anteriores.

Las mejoras diseñadas para el “*subtractive*” básico que dan lugar al “*subtractive enlazado*” con identificación y validación no disminuyen, por tanto, de forma significativa el t_{exe} del clasificador, pero tampoco lo aumentan, tal y como se demuestra en este apartado.

La Figura 5.53 permite demostrar esta afirmación mediante un ejemplo. En ella se presenta el t_{exe} de ambos algoritmos (el “*subtractive*” básico en verde y el enlazado con proceso de

identificación y validación en rojo) en el experimento complejo, observándose que éste es semejante para los dos algoritmos en toda la prueba. No se incluye en la figura el resultado generado por el “*subtractive enlazado*” sin identificación ni validación, puesto que no se aprecia ninguna diferencia respecto al arrojado por el clasificador con estos dos procesos.

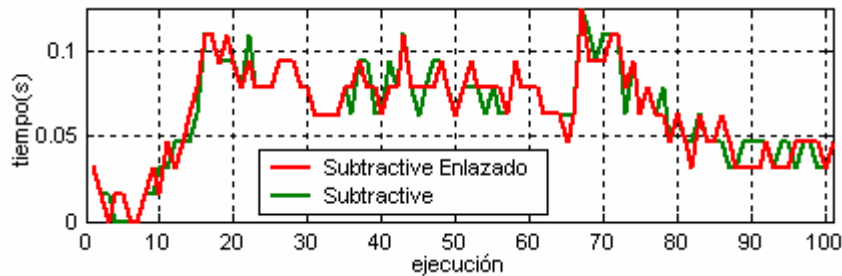


Figura 5.53. Gráfica comparativa del t_{exe} de la versión básica del “*subtractive*” y del “*subtractive enlazado*” con procesos de identificación y validación, en el experimento complejo.

Por su parte, la Figura 5.54 se muestra como cambia este tiempo en función del número de clases k_t que el algoritmo genera a su salida, al ejecutarlo en el experimento completo largo. La figura demuestra que no se existe ninguna diferencia entre las dos versiones del “*subtractive*” en lo referente a esta dependencia.

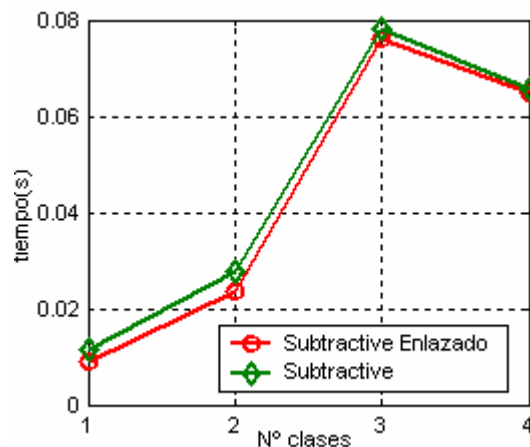


Figura 5.54. Gráfica comparativa del t_{exe} de la versión básica del “*subtractive*” y del “*subtractive enlazado*” con procesos de identificación y validación, en función del número de clases k_t que generan, en el test completo largo.

La Figura 5.54 sí que evidencia que la media del t_{exe} del “*subtractive enlazado*” con identificación y validación es algo menor a la del “*subtractive*” básico, hecho no observable en el valor instantáneo de esta variable de la Figura 5.53. Esta ligera diferencia se debe fundamentalmente al proceso de anulación de verosimilitud introducido en la versión enlazada del algoritmo, que, como se comenta en el apartado 5.3.3, simplifica el proceso de recálculo de la densidad de los puntos de Y , $p_{iter+1,i} / i = 1:m$.

El algoritmo “*subtractive*” es, al igual que el “*k-medias*”, un proceso iterativo que, sin embargo no muestra problemas de convergencia de éste último, pues el número de veces que

itera el primero para resolver la clasificación está determinado por k_t , tal y como se extrae de su funcionalidad. Debido a ello, no tiene interés analizar el efecto de esta otra variable en el t_{exe} , una vez estudiada la Figura 5.54.

Finalmente, conviene destacar el pico que aparece en las dos gráficas de la Figura 5.54, que a simple vista parece significar que el t_{exe} de las dos versiones del “subtractive” es mayor cuando generan tres clases que cuando generan dos. Evidentemente, la causa de este pico no es ésta, sino el alto número de datos a clasificar m_t que aparece durante gran parte del experimento (al principio, representado en la Figura 5.52, y al final de éste, cuando los objetos están próximos a las cámaras) asociados a tres objetos.

De este análisis se extrae la conclusión de que el t_{exe} de las versiones del “subtractive” es fuertemente dependiente de m_t , como consecuencia de la complejidad computacional del clasificador, $O(m^2)$. En el apartado 5.4.3 posterior se demuestra esta afirmación.

c) Distorsión y separación de las clases:

La Figura 5.55 muestra el valor de separación (gráfica superior) y distorsión (gráfica inferior) arrojado en el experimento complejo por el “subtractive” en sus versiones básica (en color verde solo en la gráfica de separación), enlazada (en rojo en ambas gráficas) y enlazada con identificación y validación (en amarillo en ambas gráficas). El valor de distorsión del “subtractive” básico no puede calcularse, al ser éste un algoritmo no enlazado.

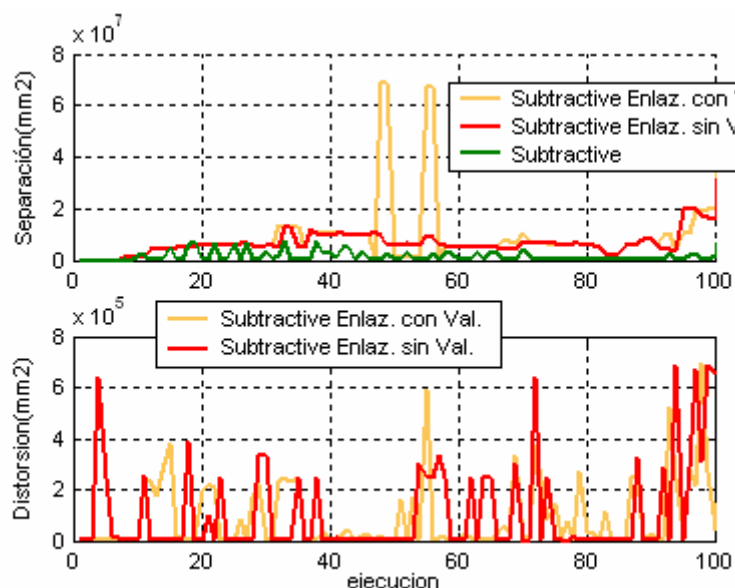


Figura 5.55. Gráficas comparativas de los valores de separación y distorsión que arroja el “subtractive” en sus distintas versiones, al ser ejecutados en el experimento complejo.

La Figura 5.55 muestra que el “subtractive enlazado”, gracias a los procesos de enlazado y anulación de verosimilitud que incluye, consigue establecer una mayor separación entre las clases que genera que la versión básica del clasificador.

Los procesos de identificación y validación aumentan aún más el valor de este factor de calidad generado por el clasificador ante el mismo experimento. Esta mejora se observa, fundamentalmente, en situaciones complejas de clasificación como la que se produce alrededor de la ejecución 55 del experimento, debida a la aparición de un objeto que oculta parcialmente a otro ya existente en la escena de interés. En circunstancias como ésta, la demostrada fiabilidad que la validación aporta al segmentador elimina el error de unión de 2 clases en 1, y permite obtener una mayor separación de las clases generadas. Este efecto es equivalente al que tiene la validación en el valor de separación que arroja el “*k-medias secuencial*”, analizado en el apartado 5.2.6 anterior.

Por otro lado, es difícil apreciar en la Figura 5.55 una diferencia importante en el valor de distorsión que generan los algoritmos “*subtractive enlazado*” con y sin validación, si bien la gráfica evidencia que el resultado arrojado por el segmentador con validación es, de media, algo menor que el del mismo sin validación. Por otro lado, las gráficas de la figura muestran un mayor valor sostenido de distorsión para el “*subtractive enlazado*” sin validación que para el mismo con validación, en la situación de oclusión analizada en el párrafo anterior, y que se debe a las razones ya expuestas.

Todos estos análisis permiten concluir que el comportamiento del clasificador borroso mejora también desde el punto de vista de los factores de calidad analizados, al introducir las propuestas descritas en el flujograma del segmentador básico, que dan lugar al “*subtractive enlazado*” con proceso de identificación y validación.

5.4. Resultados comparativos de los clasificadores propuestos: “*k-medias secuencial*” y “*subtractive enlazado*”

Una vez expuesto y analizado el funcionamiento del “*k-medias secuencial*” y del “*subtractive enlazado*” propuestos, en este apartado se compara su comportamiento para conocer qué alternativa es más interesante en la aplicación de MTT de interés en la tesis. A partir de las conclusiones extraídas en este apartado, en el capítulo 6 se describe y examina la incorporación del proceso de segmentación más adecuado en cada etapa del estimador multimodal basado en el PF, para obtener el algoritmo XPFCP propuesto en la tesis.

Para comparar el comportamiento de las dos propuestas de clasificación se utilizan las mismas variables usadas en análisis anteriores, pues permiten estudiar el cumplimiento de las especificaciones establecidas en el capítulo 1. Se incluye aquí, por tanto, una comparativa en términos de exactitud, tiempo y robustez.

5.4.1. Fiabilidad, tasa de error, y valores de distorsión y separación

Con objeto de realizar un análisis de fiabilidad de las dos propuestas de clasificación presentadas en los puntos anteriores, en la Figura 5.56 se unifican los resultados de fiabilidad del “*k-medias secuencial*” (en tonos azules) y del “*subtractive enlazado*” (en tonos verdes), ambos con validación, mostrados respectivamente en la Figura 5.23 y en la Figura 5.48. Varias conclusiones finales se extraen a la vista de la Figura 5.56:

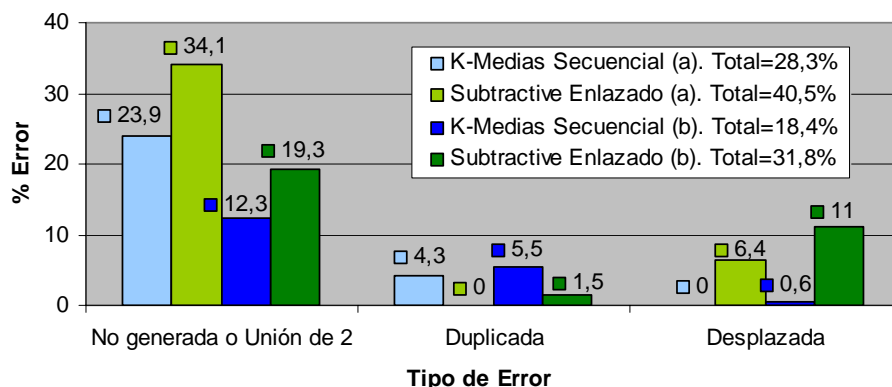


Figura 5.56. Tasa de error de los algoritmos “*k-medias secuencial*” con validación (en tonos azules) y “*subtractive enlazado*” con identificación y validación (en tonos verdes) en los experimentos complejo (a) y completo corto (b).

- La propuesta “*k-medias secuencial*” con validación tiene una menor tasa de error que la propuesta “*subtractive enlazado*” con identificación y validación, en los dos experimentos mostrados. En el experimento completo, la tasa total de error del primero queda por debajo del 20% (exactamente un 18.4%), mientras que la del segundo está algo por encima del 30% (alcanza el 31.8%). La complicación del experimento incrementa ambas tasas de error casi por igual en un 10%, pues el segmentador booleano produce un error total del 28.3%, y el borroso uno del 40.5%. Hay que remarcar que los resultados obtenidos en el experimento complejo no son extrapolables a un funcionamiento normal del clasificador, pues las condiciones que se presentan en este test han sido especialmente elegidas para probar el comportamiento de los diferentes algoritmos en situaciones límite.
- Analizando de forma individual los distintos tipos de error que producen los segmentadores en los dos tests, se observa que el error más recurrente es el más importante en la aplicación de MTT de interés: el de clases no generadas o unión de 2 en 1. En el experimento completo corto, la tasa de este tipo de error es de poco más del 10% (el 12.3%, exactamente) en los resultados arrojados por el clasificador booleano, mientras que en los del borroso ésta sube casi al 20% (concretamente al 19.3%). Este resultado se debe a la distinta implementación del proceso de identificación (predicción de centroides) incluido en los dos segmentadores; mientras que el del “*k-medias secuencial*” es intrínseco al algoritmo, el del “*subtractive enlazado*” se añade a su funcionalidad básica. El proceso de predicción de centroides implícito en el “*k-medias secuencial*” reduce esta tasa de error en un 10% respecto a la que arroja el “*k-medias extendido*” (ver la Figura 5.22), alcanzando un valor casi igual al que genera el segmentador al añadirle la validación (en la Figura 5.56). Sin embargo, en el caso del “*subtractive*” esta tasa de error sólo se decreta en un 5% al añadir a éste las modificaciones propuestas (pasa del 24.3% en el “*subtractive*” básico al 19.3% en el “*subtractive enlazado*” con identificación y validación, como muestran la Figura 5.47 y la Figura 5.48), demostrándose que en este caso la predicción de centroides no tiene el mismo efecto positivo que tiene en el algoritmo booleano.
- En contrapartida, el proceso de identificación implícito en el “*k-medias secuencial*” provoca errores de duplicación de clases inexistentes en el resultado generado por la versión básica del clasificador. La validación elimina muchos de estos errores, haciendo que el valor medio

de la tasa de este tipo de error quede finalmente en un 5%, como muestra la Figura 5.56. En esta figura se observa también que la tasa de error por duplicados, generada por el “*subtractive enlazado*” con identificación y validación, es prácticamente nula (con un valor medio en los dos tests del 0.75%) gracias, en este caso, al proceso de anulación de verosimilitud que incluye la versión enlazada del clasificador borroso, como se explica en el apartado 5.3.3.

- Finalmente, en lo referente a los errores por desplazamiento de clases, la Figura 5.56 muestra el resultado opuesto al descrito en el párrafo anterior. En el algoritmo “*subtractive*” este error no se ve disminuido al incorporar ninguna de las mejoras expuestas en apartados anteriores, obteniéndose un valor medio final para la tasa de este tipo de error del 8.7%. Por su parte, el “*k-medias*” no genera apenas errores de este tipo (su versión final arroja un valor medio del 0.3% para esta tasa de error). La justificación de esta diferencia se encuentra en la funcionalidad de cada clasificador. El “*subtractive*” elige como \bar{g}_j de cada clase uno de los datos a clasificar, que puede o no representar a la media aritmética de los que finalmente se asocian a ella. Sin embargo en el “*k-medias*” el centroide que el algoritmo genera para cada clase es la media aritmética de sus datos asociados. Las clases, centradas en sus centroides, pueden aparecer más desplazadas en los resultados generados por el clasificador borroso que en los que genera el booleano.

De la información expuesta en los párrafos anteriores se extraen las siguientes conclusiones finales sobre la fiabilidad de los algoritmos de clasificación descritos en el capítulo:

1. El “*k-medias secuencial*” con validación tiene de media una tasa global de error del 23.3%, 13.2 puntos menos que la arrojada por el “*subtractive enlazado*” con identificación y validación. El mejor funcionamiento del segmentador booleano frente al borroso se debe al carácter intrínseco del proceso de predicción de centroides en él, inexistente en el segundo.
2. Esta tasa de error global se distribuye, entre los distintos tipos de error, de forma parecida en los dos segmentadores propuestos, salvando las diferencias funcionales existentes entre éstos: alrededor del 77% de ésta es debida a los errores de no generación de clases producidos por ambos algoritmos, mientras que el resto se debe a su duplicación, en el caso del “*k-medias secuencial*”, y a su desplazamiento, en el caso del “*subtractive enlazado*”.
3. A partir de esta información se evidencia que el clasificador booleano propuesto es más fiable que el borroso, pues genera menos errores en general y de no generación de clases y unión de 2 en 1 en particular (el tipo de error más importante en la aplicación de interés, tal y como se explica en el apéndice III).
4. Una razón de la mayor fiabilidad del algoritmo booleano frente a la del borroso, en los experimentos mostrados, es su mejor comportamiento frente a conjuntos de datos de densidad desigual y variable entre las distintas clases (propiedad típica, como ya se ha comentado, de los sets de medidas en la aplicación de MTT de interés). Este hecho se debe que la base funcional del “*k-medias*” es la minimización del parámetro de distorsión (ver en el apéndice II), lo que conlleva que el algoritmo realiza una mejor distribución de los datos entre las clases, compensando el problema comentado. Se puede concluir, por tanto, que el clasificador borroso es más sensible a los problemas de sensado en la aplicación de interés.

El “*k-medias secuencial*” con validación parece, por tanto, más adecuado para realizar la clasificación de medidas necesaria en el proceso de estimación multimodal, como se expone en el capítulo 3. Por su parte, el “*subtractive enlazado*” con identificación y validación puede tener un funcionamiento igual de adecuado que el anterior en la clasificación de partículas a la salida de este proceso de estimación, pues este conjunto no presenta el problema de distribución descrito. En el capítulo siguiente se analiza en detalle los efectos de estas asignaciones en la eficiencia del XPFCP.

La Figura 5.57 muestra un resumen de la tasa de ruido no filtrado obtenida de la ejecución en el experimento de ruido de los dos clasificadores en su versión básica (“*Algoritmo Básico*” en azul y verde en la figura), en su versión secuencial (para el “*k-medias*”) o enlazada (para el “*subtractive*”) (“*Algoritmo Mejorado*” en rojo en la figura), y en su versión validada (“*Algoritmo Mejorado con Validación*” en amarillo en la figura).

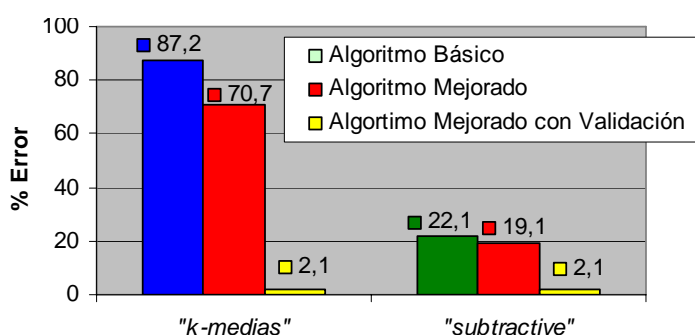


Figura 5.57. Tasa de ruido no filtrado de las distintas versiones del “*k-medias*” y del “*subtractive*” en el experimento de ruido.

La conclusión fundamental que se extrae de la Figura 5.57 es la alta e idéntica robustez frente al ruido de la versión final de las dos propuestas de clasificación diseñadas: la tasa de rechazo al ruido alcanza el 97.9% en ambos casos.

También se observa en la Figura 5.57 que el elemento fundamental causante de esta robustez es el proceso de validación (su incorporación al segmentador mejora la tasa de rechazo del algoritmo booleano en casi un 70% y la del borroso en un 17%). Las modificaciones funcionales propuestas en la tesis a la versión básica de los dos clasificadores tienen un efecto mínimo en lo que se refiere al robustecimiento del algoritmo frente al ruido: la versión enlazada del “*subtractive*” mejora en un 3% este comportamiento frente a la versión básica, mientras que la mejora del 17% que aporta la versión secuencial del “*k-medias*”, no es sino una consecuencia secundaria de la ubicación de los ruidos en el experimento de prueba, como se explica en el apartado 5.2.6 anterior.

Finalmente, destacar que se confirma el mejor comportamiento de las distintas versiones del “*subtractive*” en el filtrado del ruido (la versión básica alcanza una tasa de filtrado del 81.9%), frente al de las versiones del “*k-medias*” (la tasa de filtrado de la versión básica es en este caso de solo el 29.3%), como se plantea al principio de este capítulo, por lo que, desde este punto de vista, el efecto de la validación es más importante en el clasificador booleano que en el borroso.

Los análisis anteriores permiten concluir que el efecto más importante del proceso de validación es mejorar el comportamiento de las dos propuestas de clasificación frente a outliers. Al margen de este efecto, solo es relevante la insensibilización frente al valor de los parámetros que consigue la validación en ambos clasificadores, pues el efecto de este proceso en la fiabilidad general de los dos segmentadores es pequeño.

En lo que a la fiabilidad en la identificación se refiere, las dos propuestas de clasificación analizadas tienen un buen comportamiento, pues presentan una tasa de error de identificación de alrededor del 0.5% (ver apartados 5.2.6 y 5.3.5), gracias al proceso de validación, que disminuye esta tasa en un 8% en el caso del “*k-medias secuencial*” y en un 10% en el del “*subtractive enlazado*” con identificación.

La Figura 5.58, que aúna los resultados de identificación obtenidos con el “*k-medias secuencial*” con validación (mostrados en la Figura 5.26), y del “*subtractive enlazado*” con identificación y validación (de la Figura 5.52), ratifica la conclusión anterior, y permite además observar que los centroides generados por el clasificador borroso están más dispersos (lo que confirma la mayor tasa de error por desplazamiento de las clases generadas por este segmentador, comentada en párrafos anteriores) que los generados por el booleano. Este hecho se debe a la ya explicada diferente definición de $\bar{g}_j / j = 1:k$ de cada uno de los clasificadores.

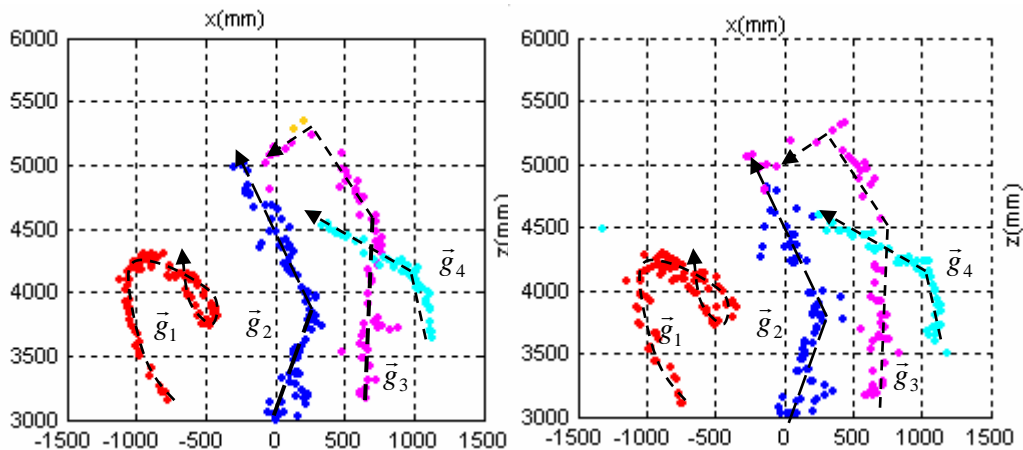


Figura 5.58. Representación por colores de la posición de $\bar{g}_{1:4}$, proyectada en el plano XZ , de cada uno de los objetos identificados por el “*k-medias secuencial*” con validación (imagen izquierda) y por el “*subtractive enlazado*” con identificación y validación (imagen derecha). Los resultados se han obtenido ejecutando los algoritmos en la segunda parte del experimento complejo. Cerca del final del experimento el objeto asociado a G_3 oculta totalmente al asociado a G_4 , pues este último pasa antes por el punto de cruce entre las dos clases.

Finalmente, en la Figura 5.59 (extraída de la Figura 5.55 y de la Figura 5.30) se muestran los valores de separación y distorsión generados por los dos clasificadores propuestos al ejecutarse en el experimento complejo. La Figura 5.59 evidencia que la separación entre las clases generada por el “*k-medias secuencial*” con validación (en azul) es mayor que la generada por el “*subtractive enlazado*” con identificación y validación (en verde), sobre todo en las situaciones complejas de clasificación debidas a oclusiones que aparecen en este experimento alrededor de la ejecución 55 (tal y como se comenta en apartados anteriores).

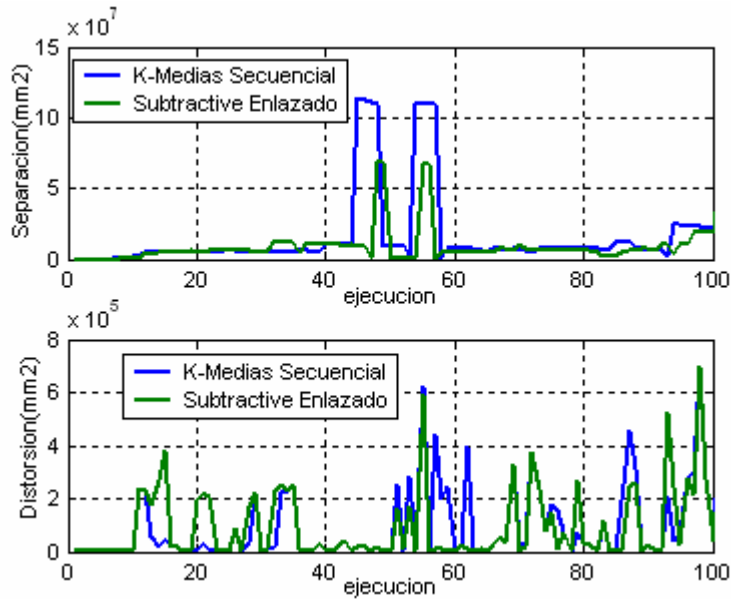


Figura 5.59. Gráficas de los valores de separación y distorsión que arroja el “*k-medias secuencial*” con validación y el “*subtractive enlazado*” con identificación y validación, al ser ejecutados en el experimento complejo.

Al margen de este análisis, es difícil apreciar en la Figura 5.59 una diferencia destacable en el valor que generan los dos segmentadores para estos factores de calidad. Debido a ello, se ejecutan de nuevo ambos en el experimento completo largo y se obtiene la diferencia normalizada (valor de cada parámetro generado por la versión propuesta del “*k-medias*”, menos el generado por la del “*subtractive*”, o “*(K-M)-(S)*” en la figura) de la separación (gráfica azul izquierda de la Figura 5.60) y de la distorsión (gráfica azul derecha de la Figura 5.60), así como la media de los valores de diferencia no nulos (en rojo en ambas gráficas).

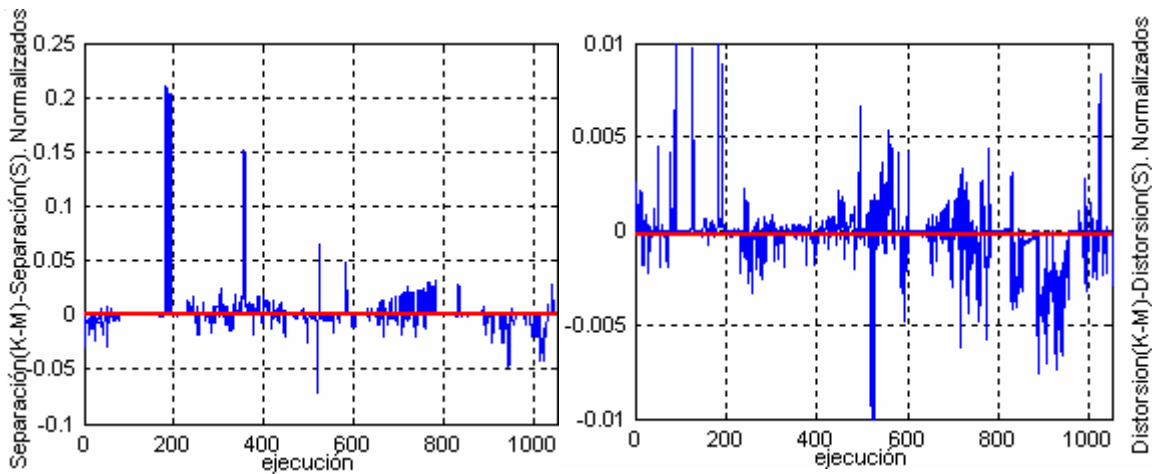


Figura 5.60. Gráficas de los valores de separación y distorsión que arroja el “*k-medias secuencial*” con validación y el “*subtractive enlazado*” con identificación y validación, al ser ejecutados en el experimento complejo.

De la Figura 5.60 resultante sí es posible concluir que, en promedio, la separación que genera el clasificador booleano entre las clases es mayor que la generada por el clasificador borroso (concretamente $0.001mm^2$ normalizados) y que en lo referente a la distorsión de las clases se produce la situación contraria (la generada por el segmentador booleano es, en promedio, $0.0002mm^2$ normalizados menor que la del borroso).

Estos resultados se deben al hecho ya comentado de que la funcionalidad del “*k-medias*” se obtiene de la optimización del parámetro de dispersión de las clases, y por ende, del de la distorsión de la clasificación, y evidencian, de nuevo, un mejor comportamiento de la propuesta booleana de clasificación frente a la borrosa.

Desde el punto de vista de la robustez frente al ruido, la fiabilidad en la identificación y los parámetros cualitativos de distorsión y separación arrojados por los clasificadores, las dos propuestas presentadas en este capítulo son, por tanto, igualmente válidas para su aplicación a las distintas etapas del XPFCP.

5.4.2. Flexibilidad, robustez frente a variaciones de los parámetros

La Figura 5.61 aúna los resultados de sensibilidad frente a los parámetros del proceso de validación aplicado al “*k-medias secuencial*” y al “*subtractive enlazado*”, mostrados en la Figura 5.25 y en la Figura 5.51 anteriores, obtenidos al ejecutar el test complejo. La conclusión fundamental que se extrae de la Figura 5.61 es la comentada en apartados anteriores: en ambas aplicaciones, la desviación típica de los errores generados por los clasificadores es pequeña al modificar de 0 a 1 el valor normalizado de los cuatro parámetros, es decir, que estos algoritmos no son muy sensibles a estos valores.

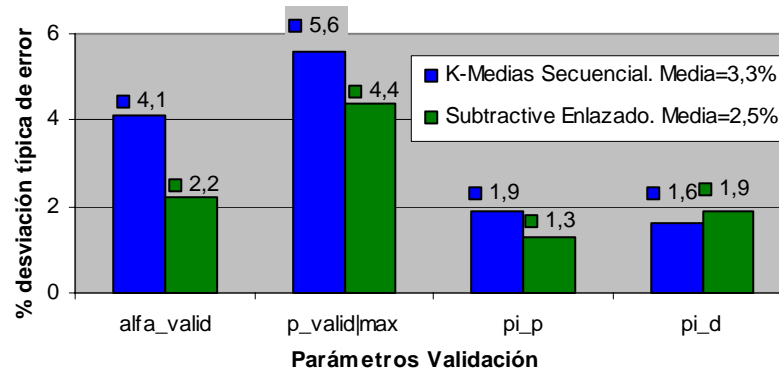


Figura 5.61. Desviación respecto a la media de error del “*subtractive enlazado*” con identificación y validación en el test complejo para los parámetros: α_{valid} (alfa_valid), $p_{valid|max}$ (p_valid|max), π_p (pi_p) y π_d (pi_d).

En la Figura 5.61 se observa también que el clasificador borroso es menos sensible (desviación media de la tasa de error del 2.5%) que el booleano (desviación media del 3.3%) al valor de todos los parámetros de validación. Esta mayor robustez se observa sobre todo en los parámetros de validación por probabilidad ($p_{valid|max}$ y α_{valid}), y se debe a que la verosimilitud de cada clase, al coincidir con la de su centroide ($p_{j,t} = p_{max|iter,t}$) como se explica en el punto 5.3.4, es filtrada por la clasificación, por lo que el filtro de probabilidad tiene un efecto menor en el algoritmo borroso que el que tiene en el booleano, cuya funcionalidad no está relacionada con la verosimilitud de las clases.

En lo referente a los parámetros de validación no normalizados, la sensibilidad de los dos clasificadores frente al parámetro β_{valid} es también equivalente (ver apartados 5.2.6 y 5.3.5). El análisis de los resultados generados al modificar de 1 a 5 este valor, arroja una desviación típica

de la tasa de error del 1.5%, ejecutando el “*k-medias secuencial*” en el experimento complejo, y del 1.9% ejecutando el “*subtractive enlazado*”¹¹.

Se puede concluir, por tanto, que ambos clasificadores son bastante insensibles a las modificaciones de los distintos parámetros del proceso de validación.

Por otro lado, si bien los parámetros de clasificación de los dos segmentadores no pueden compararse directamente, pues no son los mismos, es posible establecer una relación a través de su funcionalidad entre el parámetro $distM$ del “*k-medias secuencial*” y $distA$ del “*subtractive enlazado*”, con objeto de comparar su robustez frente a la modificación de su valor.

Al realizar este análisis se concluye que la versión básica del segmentador borroso es algo menos sensible a la modificación de $distA$ de lo que la versión básica del booleano lo es de $distM$. Esta conclusión se extrae comparando los resultados de sensibilidad expuestos en la Figura 5.9 y en la Figura 5.36, donde se observa que una modificación del 100% sobre el valor de ajuste de $distA$ genera una desviación típica máxima en la tasa de error del “*subtractive*” del 20.3%, mientras que ésta alcanza el 23.9% si se realiza una modificación idéntica en las mismas condiciones de prueba sobre el parámetro $distM$ del “*k-medias*”.

Al realizar la misma prueba sobre la versión con validación de los clasificadores se observa (ver la Figura 5.24 y la Figura 5.49) que en ambos casos la tasa de error generada por ellos se estabiliza para un rango determinado del parámetro, por lo que puede concluirse que los dos segmentadores son igualmente insensibles a los parámetros de clasificación.

5.4.3. Tiempo de ejecución

Como se comenta en apartados anteriores tanto el “*k-medias*” como el “*subtractive*” son procesos iterativos, en cualquiera de sus versiones, con idéntica complejidad computacional $O(m^2)$. En el segundo el número de veces que itera antes de generar una solución está determinado por el número de clases generadas k_i , mientras que en el primero este número es generalmente inferior al de clases (ver la Figura 5.29). Por otro lado, el clasificador booleano puede sufrir problemas de convergencia y ver entonces aumentado en gran medida su t_{exe} , situación que no se produce con el clasificador borroso. Sin embargo, tal y como demuestra la Figura 5.29, la versión secuencial del “*k-medias*” mejora la convergencia del algoritmo básico, disminuyendo el número de iteraciones necesarias para generar su salida.

En general, y en cualquier versión de los algoritmos, el t_{exe} del “*subtractive*” es mayor que el del “*k-medias*” pues, debido a su funcionalidad básica, el tiempo de cálculo asociado a cada \bar{y}_i a clasificar es menor en el caso del algoritmo booleano que en el del borroso.

¹¹ El ajuste de los otros dos parámetros de validación no normalizados es teórico (como el de k_{valid}) o depende del ajuste de otros parámetros de clasificación (como el de d_{valid}): $distM$ (en el caso del “*k-medias secuencial*”), o $distA$ y $distB$ (en el caso del “*subtractive enlazado*”), como se comenta en apartado anteriores.

La Figura 5.62, que corrobora estos análisis, muestra el t_{exe} del “*k-medias secuencial*” con validación (en azul) frente al del “*subtractive enlazado*” con identificación y validación (en verde), al ser ejecutados en el experimento complejo. La figura reúne los resultados mostrados en las anteriores Figura 5.53 y Figura 5.28.

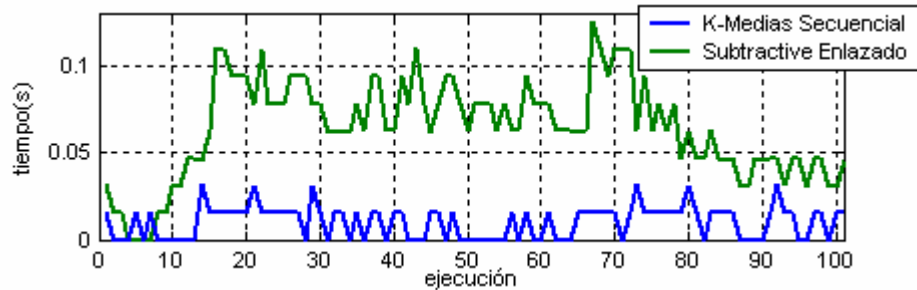


Figura 5.62. Gráfica comparativa del t_{exe} del “*k-medias secuencial*” con validación y del “*subtractive enlazado*” con identificación y validación, en el experimento complejo.

Para corroborar los resultados mostrados en la Figura 5.62, se ejecutan los algoritmos en otros experimentos, resultando siempre que el t_{exe} del algoritmo borroso es semejante al del booleano en las situaciones sencillas de clasificación en las que hay pocos datos, asociados a una o como máximo dos clases (como al principio del experimento complejo, en la figura) es doble al del booleano en situaciones con una cantidad media de datos asociados a 2 o más clases (como al final del experimento complejo), y es aún mayor cuando el número de datos m_i aumenta en gran medida o las clasificación se complica (por ejemplo desde la ejecución 15 a la 80 en el experimento complejo).

Por su parte, en la Figura 5.63 incluye dos gráficas comparativas del efecto que tienen tanto k_i (gráfica izquierda) como m_i (gráfica derecha), en el t_{exe} del “*k-medias secuencial*” con validación (en azul) y del “*subtractive enlazado*” con identificación y validación (en verde).

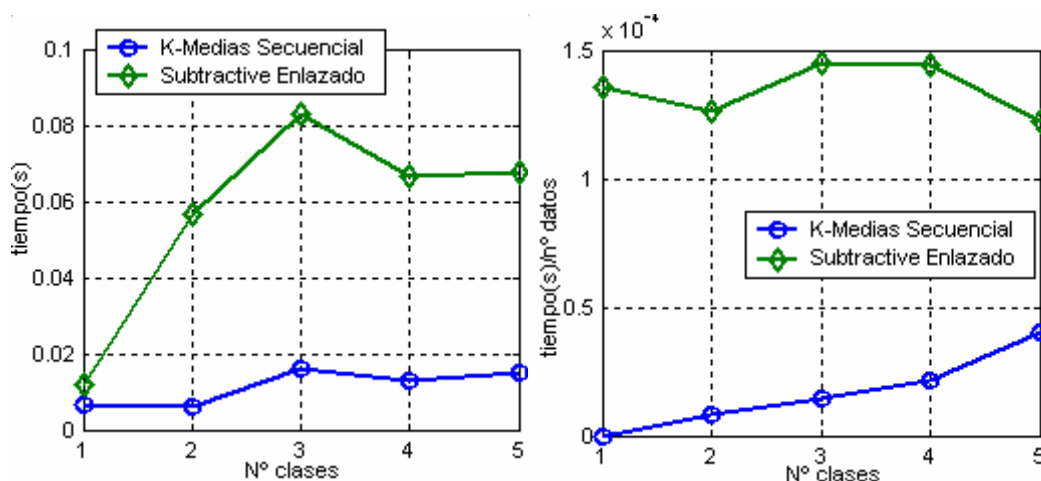


Figura 5.63. Gráficas comparativas del t_{exe} del “*k-medias secuencial*” con validación y del “*subtractive enlazado*” con identificación y validación, en función del número de clases que arrojan como salida (gráfica izquierda), y de éste y el número de datos a clasificar (gráfica derecha) en el test completo largo.

La gráfica izquierda de la Figura 5.63 aún los resultados mostrados en las anteriores Figura 5.29 y Figura 5.54. En ella se aprecia la dependencia del t_{exe} de los dos clasificadores de k_t , y que este tiempo es en el algoritmo booleano, en todos los casos, menor que el del borroso.

En la gráfica derecha de la Figura 5.63 se muestra la misma dependencia comentada en el párrafo anterior, pero esta vez ponderada por el valor de m_t en cada instante. El resultado indica que, como se comenta anteriormente, el t_{exe} en el “*k-medias*” depende sobre todo de k_t , y es casi independiente de m_t , mientras que con el “*subtractive*” se produce la situación opuesta.

5.5. Conclusiones y resumen de las aportaciones realizadas en esta área

En este capítulo se describen dos propuestas de clasificación determinísticas (el “*k-medias secuencial*” con validación y el “*subtractive enlazado*” con identificación y validación), adaptadas para resolver la tarea de MTT de interés en la tesis, en combinación con un algoritmo de estimación probabilístico basado en el PF (el XPFCP), tal y como se expone en el capítulo 3.

Las dos funciones a desarrollar por los procesos de segmentación aquí propuestos y descritos en el XPFCP, se analizan en dicho capítulo 3 y se resumen a continuación:

1. Como clasificador del conjunto de medidas de posición Y , extraídas de los distintos objetos implicados en la tarea de seguimiento, el objetivo a cumplir por el segmentador es el de robustecer el comportamiento del estimador bayesiano, incrementado el peso de las partículas que caracterizan a las hipótesis más pobremente sensadas, de modo que tengan una representación suficientemente importante en la creencia a la salida del filtro como para no ser eliminadas por la degeneración de ésta. El conjunto de clases generado por los clasificadores se usa además para que el seguidor incorpore y elimine automáticamente del proceso de estimación, las distintas hipótesis de objetos a seguir según éstos aparecen y desaparecen, respectivamente, del entorno bajo estudio en cada instante.
2. Como clasificador del conjunto de partículas que caracterizan a la creencia, a la salida del XPFCP, el objetivo a cumplir por el segmentador es obtener una salida determinística que indique la identidad, posición y velocidad estimada de cada objeto, y un valor de probabilidad que indique de la fiabilidad de la estimación obtenida.

El modo y la demostración de que estos objetivos se consiguen mediante la combinación de los clasificadores propuestos y el estimador bayesiano, se exponen en los capítulos siguientes. Las principales aportaciones y conclusiones que se extraen sobre cada uno de los algoritmos descritos en este capítulo se resumen a continuación:

- El “*k-medias secuencial*” se obtiene modificando el estándar “*k-medias*”, de modo que el algoritmo resultante sea más eficiente en la búsqueda de los distintos conjuntos de datos en la matriz de entrada Y , y por ende, se mejore la convergencia de este algoritmo iterativo y se disminuya su t_{exe} , sobre todo en situaciones complejas (el t_{exe} de la versión modificada se decrementa hasta en un 75% respecto al de la versión básica).

Además, el segmentador así obtenido incluye de forma implícita un proceso de asociación entre cada clase y el modelo de movimiento del objeto del que se supone proceden las medidas de posición incluidas en la clase, que permite identificar a lo largo del tiempo a cada clase de medidas, y por ende, al objeto de las que se extraen en la aplicación de MTT.

La fiabilidad del “*k-medias secuencial*” mejora gracias a esta modificación frente a la del algoritmo básico: la tasa de error de clases no generadas (el más importante según se comenta en el apéndice III) disminuye de media en un 10%, si bien se genera, en contrapartida, errores por duplicación de clases (este error alcanza una tasa del 5% en la versión final del clasificador). Este último hecho no es un factor importante en la elección del algoritmo como clasificador de medidas en el XPFCP, frente a su bajo t_{exe} y su fiabilidad, que lo hacen especialmente adecuado para esta tarea, como ya se ha comentado y se demuestra en el capítulo 6. La fiabilidad del clasificador borroso propuesto en este capítulo es un 13% menor que la del booleano (tasa de error de la que 8.5 puntos se deben a errores de no generación de clases); este hecho se debe a su peor comportamiento ante conjuntos de datos desigualmente distribuidos entre las clases (ver apartado 5.4.1 anterior), lo que desaconseja su uso como clasificador de medidas en el XPFCP.

Se añade finalmente al clasificador una etapa de validación que depende de la verosimilitud del conjunto de medidas que reúne cada clase y de la exactitud conseguida en la predicción del valor del centroide, calculada en el mencionado proceso de asociación. La validación incrementa la fiabilidad del proceso de identificación implicado en la asociación (la tasa de aciertos de identificación pasa del 91.5% que genera el proceso sin validación al 99.4% del mismo con validación).

Por otro lado, esta etapa de validación robustece la salida del clasificador booleano, aumentando su tasa de rechazo al ruido (pasa del 29.3% sin proceso de validación al 97.9% con él) y disminuyendo su sensibilidad respecto a modificaciones de su parámetro básico $distM$ (se consigue una tasa de error estable a partir de un valor determinado de éste).

- El “*subtractive enlazado*” es una versión con conexión (enlazada) del algoritmo de clasificación borroso “*subtractive*” ([Yager94]). Teniendo en cuenta que los miembros de cada clase son necesarios en el paso de re-inicialización del XPFCP (como se expone en capítulos anteriores) es necesario añadir a la versión básica del clasificador este proceso de enlazado de datos a cada clase.

Por otro lado, además del proceso de enlazado descrito en el párrafo anterior, y a partir de él, se incluye también en el “*subtractive enlazado*” una etapa de anulación de verosimilitud de los datos enlazados a cada clase. Esta etapa elimina el (aproximadamente) 5% de errores por duplicación de clases generados por el algoritmo, haciéndolo así más interesante que el clasificador booleano para la tarea de clasificación del set de partículas a la salida del XPFCP, tal y como se comenta en el punto 5.4.1 anterior.

La robustez del algoritmo así obtenido se mejora además añadiéndole el proceso de validación también incluido en el “*k-medias secuencial*”, de modo que la tasa de rechazo al ruido del “*subtractive enlazado*” con validación alcanza el valor generado por el clasificador

booleano (97.9%). El proceso de validación tiene también el mismo efecto que en el “*k-medias secuencial*”, de insensibilizar la fiabilidad del clasificador respecto a la modificación del valor de sus parámetros (fundamentalmente $disA$ y α_{new}).

El proceso de validación junto al de anulación de la verosimilitud comentado, mejora también la fiabilidad del clasificador, disminuyendo en algo más de un 3% la tasa de error que arroja la versión final del “*subtractive enlazado*” debida a clases no generadas, respecto a la de la versión básica. En contrapartida, la versión final del clasificador aumenta la tasa de error por desplazamiento de clases, respecto a la que arroja el “*subtractive*” básico, en alrededor del 2% (alcanzado un valor medio de casi el 9%), lo cual es un inconveniente en su aplicación en el XPFCP a la tarea mencionada. En el capítulo 6 se analiza esta cuestión.

Para poder incorporar el proceso de validación al “*subtractive enlazado*”, del mismo modo que se hace en el clasificador booleano, es necesario añadirle previamente una etapa de identificación, innecesaria en el caso del clasificador booleano por estar implícita en él. El análisis de fiabilidad de este último proceso indica que la validación aumenta la tasa de aciertos de identificación en alrededor de un 10% (algo más que en el clasificador booleano), dejándola en un valor casi idéntico al que arroja el “*k-medias secuencial*” con validación.

El t_{exe} de la versión final del clasificador borroso es siempre superior al de la versión final del clasificador booleano (de media, entre 4 y 8 veces superior, dependiendo del número de clases generadas k_t en el caso del clasificador booleano, y del número de datos a clasificar m_t en el caso del borroso). El t_{exe} de la versión final del “*subtractive enlazado*” no se mejora con ninguna de las modificaciones realizadas sobre el “*subtractive*” básico.

Con todo lo expuesto, se puede concluir que las propuestas de clasificación diseñadas en esta tesis mejoran en gran medida las características de fiabilidad, robustez y t_{exe} de las versiones básicas a partir de las que se generan. Además se concluye también que los segmentadores aquí presentados pueden emplearse en el algoritmo de estimación multimodal propuesto en la tesis para desarrollar con eficacia las funciones expuestas al inicio de este punto.

Con este objetivo, ambas propuestas de clasificación generan a su salida la estructura de datos relativos a cada una de las clases obtenidas $G_{j,t}$, que se muestra a continuación a modo de resumen (expuesto en las ecuaciones <5.20> y <5.41> anteriores):

$$G_{j,t} / j = 1 : k_t \equiv \{ \bar{g}_{j,t}, L_{j,t}, \tau_j, \beta_{j,t}, p_{j,t} / j = 1 : k_t \} \quad <5.44>$$

Cabe destacar que, si bien cualquier de los dos procesos de segmentación propuestos da una información aproximada inicial acerca de la posición, identificación y número de objetos de los que se extraen el conjunto de medidas a clasificar en su aplicación a la tarea de MTT de interés, ninguno de ellos genera por sí solo una estimación de estos datos suficientemente fiable en el marco de robótica personal descrito en el capítulo 1. La combinación de esta información con el proceso de estimación multimodal basado en el PF descrito en el capítulo anterior, proporciona, sin embargo, una solución al problema de seguimiento de múltiples objetos que cumple las especificaciones impuestas en la tesis, como se demuestra en el capítulo siguiente.

CAPÍTULO 6

DESCRIPCIÓN FINAL DEL SISTEMA Y RESULTADOS

En este capítulo se presenta la solución combinada propuesta en la tesis para resolver la tarea de seguimiento de múltiples objetos en entornos complejos descrita en el capítulo 1, el XPFCP, “*Filtro de Partículas Extendido con Proceso de Clasificación*”.

Tal y como se expone en el capítulo 3, el XPFCP combina la propuesta de estimación probabilística multimodal basada en el PF que se presenta en el capítulo 4 (el XPF) con la de clasificación determinística para la que se plantean una alternativa booleana (el “*k-medias secuencial*” con validación) y otra borrosa (el “*subtractive enlazado*” con identificación y validación), descritas en el capítulo 5.

En los apartados siguientes se describen cada una de las etapas del XPFCP obtenidas al incorporar el clasificador de medidas $Y_t \Rightarrow G_{1:k,t|in}$, y los resultados que éste genera $G_{j,t|in} / j = 1:k_{in,t}$, a las correspondientes del XPF. De este modo se analizan las tareas que el segmentador realiza para que el filtro probabilístico, y más concretamente, la función de creencia que éste genera como estimación del vector de seguimiento, se adapte a las especificaciones planteadas en el capítulo 1 para la aplicación de MTT.

Una vez descrita la funcionalidad del XPFCP, se realiza un análisis teórico y empírico de los resultados que con él se obtienen en la tarea de seguimiento de múltiples objetos perseguida. El

análisis incluye el efecto que tiene la modificación de los distintos parámetros del filtro multimodal en su funcionalidad. Las pruebas mostradas en los apartados de resultados permiten confirmar, además, la idoneidad de cada una de las dos propuestas de clasificación presentadas en el capítulo 5, así como de los procesos de identificación y validación que éstas incluyen, en las distintas etapas de la aplicación de MTT.

Una vez ajustado el funcionamiento del filtro, y para completar el conjunto de resultados generados por el XPFCP en la tarea de MTT de interés, se incluye también en el capítulo los obtenidos a partir de un sistema de observación diferente al usado hasta este punto para extraer el set de medidas Y_t de entrada al algoritmo. Como se adelanta en el capítulo 1, este otro sistema de medida está basado en un conjunto de cámaras estáticas ubicadas en una configuración de “*Espacio Inteligente*”.

El análisis de resultados se hace en base a los experimentos y criterios de fiabilidad, tiempo y robustez empleados en el capítulo 5. Las explicaciones incluidas en el apéndice III son, por tanto, referenciadas también en este capítulo, para definir las pruebas y análisis expuestos.

Finalmente, se incluye también en este capítulo un análisis comparativo de la funcionalidad del XPFCP, frente a otros algoritmos de MTT en los mismos términos de fiabilidad, sensibilidad, robustez y t_{exe} . Se realiza una comparativa de los resultados obtenidos con el algoritmo propuesto frente al que sirve como punto de partida de éste: el XPF; y frente a otros dos típicamente empleados por la comunidad científica para resolver la tarea de seguimiento de interés: el JPDAF y el S-JPDAF.

Los resultados expuestos en este capítulo son, por tanto, la confirmación funcional de las aportaciones de esta tesis al problema de MTT planteado.

6.1. Incorporación del proceso de clasificación al “*Filtro de Partículas Extendido*” (XPF): el “*Filtro de Partículas Extendido con Proceso de Clasificación*” (XPFCP)

En este apartado se describen las modificaciones que se aportan al XPF para dar lugar a la propuesta final de estimación multimodal para la aplicación de MTT perseguida, a través de la incorporación a su flujograma del proceso de clasificación descrito en el capítulo 5.

Antes de proceder con el análisis específico de las modificaciones comentadas, conviene recordar los objetivos que se persiguen con ellas:

1. *Robustecer el funcionamiento del XPF*, de modo que los distintos máximos de la creencia multimodal $p(\bar{x}_t | \bar{y}_{1:t})$ que éste genera caractericen el comportamiento estimado de cada uno de los objetos seguidos en su aplicación a la tarea de MTT de interés.
2. *Mejorar su comportamiento*, de modo que se flexibilice y se insensibilice la eficacia de estimación frente a conjuntos de medidas Y_t ruidosos y poco densos o de densidad variable; y se evite el problema de empobrecimiento del set que caracteriza a la creencia de MTT.

3. *Resolver de forma fiable el problema de asociación* intrínseco a la tarea de seguimiento de múltiples objetos, e insertar la solución propuesta en el estimador bayesiano, de modo que el filtro multimodal resultante pueda, además, ejecutarse en tiempo real.
4. *Generar una solución determinística*, a partir de la salida probabilística generada por el filtro bayesiano $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$, que permita saber en todo momento el número de objetos seguidos en el entorno analizado $k_{out,t}$, además de su estado.

En cada una de las modificaciones analizadas a continuación se justifica en cuáles de estas líneas se mejoran con ella el comportamiento del algoritmo resultante, y qué ventajas aporta frente a otras propuestas realizadas por la comunidad científica para alcanzar estos objetivos.

6.1.1. Descripción general del XPFCP

La Figura 6.1 y la Figura 6.2 muestran, respectivamente, el diagrama de bloques y una representación funcional (para el caso de un vector de estado $\bar{x} = x$ unidimensional) del XPFCP. Las diferencias (marcadas con fondo azul) entre éstas y la Figura 4.5 y la Figura 4.6 (en el capítulo 4), respectivamente, presentan las mejoras que la propuesta probabilística diseñada aporta al XPF a través de la inserción en su flujograma del proceso de clasificación de medidas $Y_t \Rightarrow G_{1:k,t|in}$ (con bloques azules en la Figura 6.1).

La Figura 6.1 incluye la llamada al clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$ (con bloque verde) que se añade a la salida del flujograma XPFCP con el objetivo, descrito en el capítulo 3, de obtener una solución determinística del proceso de seguimiento de múltiples objetos, que tenga en cuenta la forma multimodal de la creencia discreta generada $p(\bar{x}_t^{(1:n-n_{m,t})} | \bar{y}_{1:t})$.

Los párrafos siguientes describen de forma concreta las modificaciones que cada uno de los pasos del XPFCP presentan respecto a la versión XPF del filtro expuesta en el capítulo 4.

La incorporación del clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$ a la salida del XPFCP no se analiza en este apartado que se centra en el funcionamiento del estimador multimodal. Al ejecutarse fuera del lazo iterativo del filtro bayesiano (ver la Figura 6.1), el clasificador no modifica la salida probabilística del XPFCP, caracterizada por el set de partículas

$$S_t = \left\{ \bar{x}_t^{(i)}, \frac{1}{n-n_{m,t}} \right\}_{i=1}^{n-n_{m,t}}, \text{ por lo que su análisis se aborda más adelante en este capítulo.}$$

a) Modificaciones en el paso de “re-inicialización”

Como se explica en el capítulo 4, la función principal del paso de “re-inicialización”, añadido a la versión básica del PF en el XPF, es la de incorporar al principio de cada ejecución en t del algoritmo, nuevas hipótesis de estimación a la creencia discreta $p(\bar{x}_{t-1}^{(1:n-n_m)} | \bar{y}_{1:t-1})$, modelada por el set S_{t-1} en la ejecución anterior del filtro.

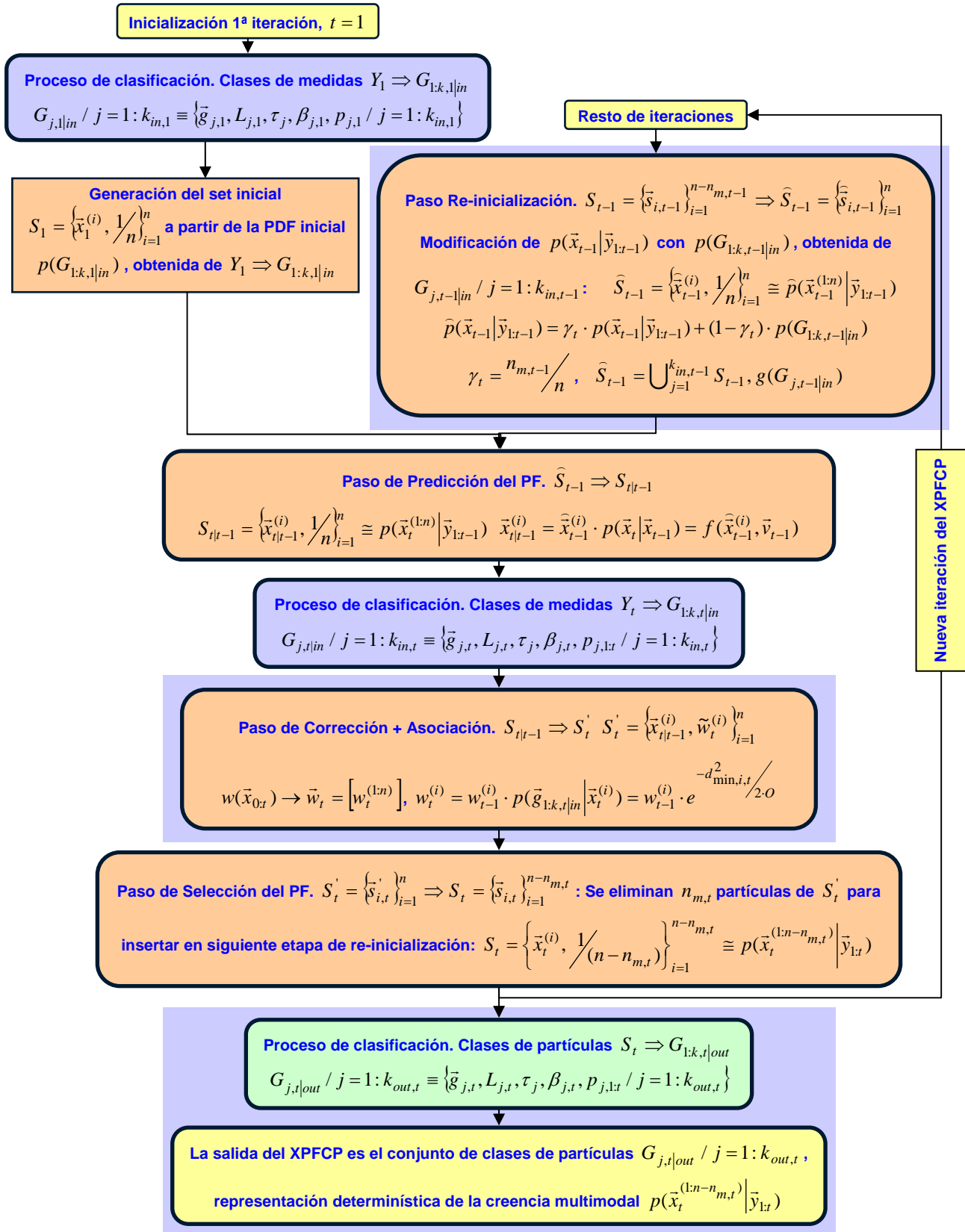


Figura 6.1. Diagrama funcional del algoritmo XPFCP. Los pasos modificados respecto al flujograma del XPF de la Figura 4.5 se muestran sobre fondo azul, así como los que implican una llamada al proceso de clasificación de medidas $G_{j,t-1|in} / j = 1 : k_{in,t-1}$. La llamada al clasificador de partículas $S_t \Rightarrow G_{j,t|out} / j = 1 : k_{out,t}$ se muestra en verde.

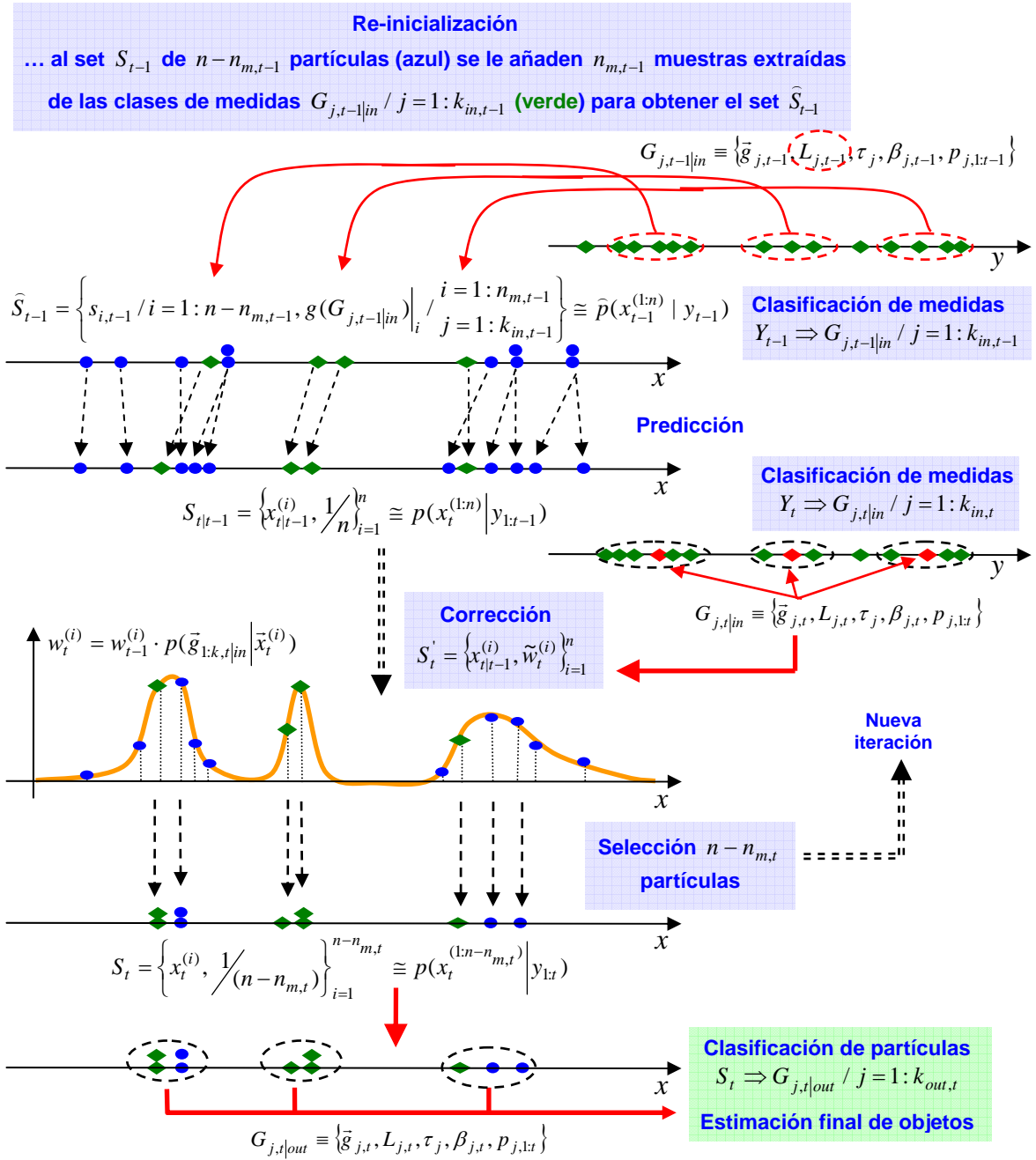


Figura 6.2. Representación unidimensional de la evolución de $S_t = \{\bar{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ en las diferentes etapas del XPFCP. Los pasos modificados respecto al flujograma del XPF de la Figura 4.6 se indican con fondo azul, y la clasificación de partículas $S_t \Rightarrow G_{j,t|out} / j = 1 : k_{out,t}$ con fondo verde. Las partículas nuevas insertadas de las clases de medidas en el paso de re-inicialización $Y_{t-1} \Rightarrow G_{j,t-1|in} / j = 1 : k_{in,t-1}$ se muestran en rombos verdes.

De este modo, en cada ejecución del XPF y antes del paso de predicción, se actualiza el conjunto de hipótesis implicadas en la estimación, con la matriz de medidas observada en la ejecución anterior Y_{t-1} (ver la Figura 4.6 en el capítulo 4).

Matemáticamente y desde el punto de vista de modificación de la creencia ($p(\bar{x}_{t-1}|\bar{y}_{1:t-1}) \Rightarrow \bar{p}(\bar{x}_{t-1}|\bar{y}_{1:t-1})$), este proceso se representa por la expresión <4.41>, mostrada en el capítulo 4, y que se repite aquí por claridad (en ella $\gamma = n_m/n$):

$$\bar{p}(\bar{x}_{t-1}|\bar{y}_{1:t-1}) = \gamma \cdot p(\bar{x}_{t-1}|\bar{y}_{1:t-1}) + (1 - \gamma) \cdot p(\bar{y}_{t-1}) \quad <6.1>$$

Sin embargo, el paso de re-inicialización así descrito en el XPF no es suficientemente robusto como para que las nuevas hipótesis introducidas en $p(\bar{x}_{t-1}|\bar{y}_{1:t-1})$ se mantengan y se refuercen en el proceso de estimación, tal y como se explica en el capítulo 4. Las mejoras que se aportan a este paso en el XPFCP para evitar este problema son las siguientes:

1. El conjunto de medidas que caracteriza a la verosimilitud $p(\bar{y}_{t-1}|\bar{x}_{t-1})$, se organiza en clases $G_{j,t-1|in} / j = 1:k_{in,t-1}$ que, gracias a la robustez del proceso de segmentación (expuesto en el capítulo 5), representan a cada una de las hipótesis de seguimiento en la aplicación de MTT de interés con una tasa de acierto del 70% (valor medio de la tasa de error global de los dos clasificadores allí analizados) y una tasa de rechazo al ruido del 97.9%. Las partículas a insertar en este paso se generan a partir de los miembros de cada clase $L_{j,t-1} / j = 1:k_{in,t-1}$, y por tanto, a partir de la PDF $p(G_{1:k,t-1|in})$ en lugar de $p(\bar{y}_{t-1})$, lo que asegura que el set $\hat{s}_{i,t-1} / i = n - n_{m,t-1} : n$ añadido al generado por el algoritmo en $t - 1$, S_{t-1} , represente a todas las hipótesis sensadas en dicho instante con la alta robustez del clasificador. El set de partículas a la salida de este paso queda modificado, por tanto, de la siguiente forma:

$$S_{t-1} = \left\{ \bar{x}_{t-1}^{(i)}, \frac{1}{n - n_{m,t-1}} \right\}_{i=1}^{n - n_{m,t-1}} \Rightarrow \bar{S}_{t-1} = \left\{ \bar{x}_{t-1}^{(i)}, \frac{1}{n} \right\}_{i=1}^n = \bigcup_{j=1}^{k_{in,t-1}} S_{t-1, j}, g(G_{j,t-1|in}) \quad <6.2>$$

2. El número de partículas a introducir en S_{t-1} en este paso, en cada ejecución del XPFCP, se varía en función de la información Y_{t-1} obtenida del proceso de observación en el instante $t - 1$. El parámetro n_m se convierte así en $n_{m,t-1}$, y el que lo relaciona con el número total de partículas del XPFCP (n) queda como sigue:

$$\gamma_t = \frac{n_{m,t-1}}{n} \quad <6.3>$$

La forma en que $n_{m,t-1}$ depende de Y_{t-1} , y por ende del tiempo, se analiza más adelante en este capítulo. Por otra parte, la nueva definición del parámetro γ_t , obliga a reescribir la ecuación <6.1>, que define matemáticamente y desde el punto de vista de la creencia, la transformación $p(\bar{x}_{t-1}|\bar{y}_{1:t-1}) \Rightarrow \bar{p}(\bar{x}_{t-1}|\bar{y}_{1:t-1})$ que se produce con este nuevo paso de re-inicialización y que coincide con la expresión <3.1> del capítulo 3:

$$\bar{p}(\bar{x}_{t-1}|\bar{y}_{1:t-1}) = \gamma_t \cdot p(\bar{x}_{t-1}|\bar{y}_{1:t-1}) + (1 - \gamma_t) \cdot p(G_{1:k,t-1|in}) \quad <6.4>$$

3. Por otro lado, las partículas $n_{m|j,t-1} / j=1:k_{in,t-1}$ a insertar en el paso de re-inicialización así descrito se obtienen del muestreo aleatorio del conjunto de miembros de cada clase de medidas $L_{j,t-1|in}$. El número de ellas se determina individualmente por cada clase, de acuerdo con el número de éstas $k_{in,t-1}$, y de la magnitud en que la hipótesis de estimación asociada a cada una de ellas está ya representada en el set S_{t-1} de entrada a este paso:

$$n_{m|j,t-1} = g(k_{in,t-1}, G_{j,t-1|in}) / j=1:k_{in,t-1} \Rightarrow n_{m,t-1} = \sum_{j=1}^{k_{in,t-1}} n_{m|j,t-1} \quad \langle 6.5 \rangle$$

La función $g()$, que define la forma de cálculo de $n_{m|j,t-1}$, se analiza más adelante.

4. Finalmente, y como se explica en el capítulo 4, y gracias al modelo de velocidad constante CV, que se propone usar en el XPFCP como se hace también en el XPF, la transformación en partículas de los miembros $\bar{y}_{i,t-1} \in L_{j,t-1|in} / j=1:k_{in,t-1}$ de las clases de medidas a insertar en la etapa de re-inicialización sólo implica añadir a sus componentes $\bar{y}_{t-1} = [x_{t-1} \ y_{t-1} \ z_{t-1}]^T$ las de velocidad $[vx_{t-1} \ vz_{t-1}]^T$, con objeto de completar el contenido del vector de estado que representa cada partícula $\bar{x}_{t-1} = [x_{t-1} \ y_{t-1} \ z_{t-1} \ vx_{t-1} \ vz_{t-1}]^T$. En la aplicación de esta transformación al XPF las componentes de velocidad de las $n_{m,t-1}$ partículas insertadas se inicializan con valor nulo, mientras que en el caso del XPFCP se usa el valor de velocidad del centroide empleado como vector de entrada $\bar{u}_{j,t-1|in} / j=1:k_{in,t-1}$ en el proceso de predicción de centroides de los clasificadores (ver en el capítulo 5):

$$\bar{u}_{j,t-1|in} = g_j(\bar{g}_{j,t-1|in}, \bar{g}_{j,t-2|in}) = [vx_{j,t-1|in} \ vz_{j,t-1|in}]^T / j=1:k_{in,t-1} \quad \langle 6.6 \rangle$$

De esta forma, las partículas insertadas $\hat{s}_{i,t-1} / i=n-n_{m,t-1}:n$ tienen, desde el momento de su incorporación al set S_{t-1} , un valor más similar al de la estimación final que con el método propuesto en el XPF. Se incrementa, por tanto, la probabilidad de permanencia al final del filtrado de estas partículas añadidas en el paso de re-inicialización, lo que es especialmente importante en las que se insertan como representación de hipótesis nuevas.

El uso de un valor de $n_{m|j,t-1} / j=1:k_{in,t-1}$ específico para cada clase permite reforzar las hipótesis sin representación en la creencia $p(\bar{x}_{t-1} | \bar{y}_{1:t-1})$ a la entrada del paso de re-inicialización, de modo que se mantengan a lo largo de todos los pasos del XPFCP, y se elimine el problema de robustez del XPF.

El ejemplo de funcionamiento del XPFCP en el instante t mostrado en la Figura 6.2 expone este efecto. En él se representa la actuación del filtro ante la aparición de un nuevo sistema a incluir en el proceso de estimación en $t-1$ (caracterizado por las medidas de valor intermedio de x en Y_{t-1}). En esta situación, el paso de re-inicialización provoca la inserción en

$p(\bar{x}_{t-1} | \bar{y}_{1:t-1})$ de $n_{m,t-1} = 4$ nuevas partículas (rombos de color verde en la figura): dos representando al nuevo modelo, y sólo una caracterizando a cada una de las dos hipótesis ya presentes en la creencia $p(\bar{x}_{t-1} | \bar{y}_{1:t-1})$ a la entrada de esta etapa (en los valores extremos de $\bar{x} = x$ y caracterizadas por los $n - n_{m,t-1} = 10$ puntos azules en la figura). En el ejemplo, por tanto, $\gamma_t = \frac{n_{m,t-1}}{n} = \frac{4}{14}$.

De esta forma, la hipótesis nueva queda representada con mayor robustez en la creencia $\hat{p}(\bar{x}_{t-1} | \bar{y}_{1:t-1})$ a la salida del paso de re-inicialización, que si la inserción de las partículas $\hat{s}_{i,t-1} / i = n - n_{m,t-1} : n$ se realiza de forma aleatoria a partir del set Y_{t-1} , como especifica la funcionalidad del XPF (ver la Figura 4.6 del capítulo 4).

Por otro lado, el uso de un valor de $n_{m|j,t-1}$ específico para cada clase de medidas $j = 1 : k_{in,t-1}$ ayuda a eliminar el problema de empobrecimiento del set S_t a la salida del XPFCP al aportarle la diversidad necesaria para evitar este efecto. El empleo de un método determinístico para elegir las partículas que se insertan en el paso de re-inicialización $\hat{s}_{i,t-1} / i = n - n_{m,t-1} : n$, asegura la renovación y la diversificación de las muestras que caracterizan todas las hipótesis en la creencia en cada ejecución del filtro.

Como en el caso del XPF, el paso de re-inicialización así diseñado tiene el mismo efecto de mover el set \hat{S}_{t-1} a su salida hacia zonas de alta verosimilitud, que los pasos MCMC o de “*Muestreo Segmentado*” propuestos por la comunidad científica para evitar el problema de empobrecimiento comentado (ver en el capítulo 4). Sin embargo, la incorporación en el XPFCP de un proceso determinístico en esta etapa, asegura además la representación de la verosimilitud de todas las hipótesis sensadas, cuestión que no se puede asegurar con el muestreo aleatorio propio del XPF. Este razonamiento demuestra la mejora que el XPFCP aporta a la resolución del problema de empobrecimiento del set que presentan el XPF y el algoritmo “*Bootstrap*”, en el que ambos basan su funcionalidad.

En apartados posteriores se demuestra, a través de un análisis de resultados generados por el XPFCP, que para conseguir el efecto de robustecimiento de las nuevas hipótesis en la re-inicialización, la opción de calcular el número de partículas a insertar en relación con cada clase $n_{m|j,t-1} / j = 1 : k_{in,t-1}$, y por ende con cada hipótesis, es mejor que la de incrementar el número de partículas total n del filtro, o el valor del parámetro de relación γ_t .

b) Modificaciones en el paso de predicción

La funcionalidad del paso de predicción en cualquier versión del PF (y en cualquier otro estimador) es la de propagar el valor de las partículas a la entrada de la etapa $\hat{x}_{t-1}^{(1:n)}$ a través del modelo de actuación $p(\bar{x}_t | \bar{x}_{t-1}) \cong f(\bar{x}_{t-1}^{(1:n)}, \bar{v}_{t-1})$ del sistema implicado en la estimación, manteniendo el valor del peso asociado a cada una de ellas $\bar{w}_t = \left[\frac{1}{n} \right]_{i=1}^n$. Esta funcionalidad no

se modifica en el XPFCP respecto a la que tiene en el XPF, por lo que lo expuesto en referencia a este paso en el punto 4.3.1 del capítulo 4 es directamente aplicable al XPFCP aquí analizado.

Más concretamente, el modelo CV completo allí expuesto (de actuación en la ecuación <4.39> y de observación en la <4.40>) es el usado también en el XPFCP para predecir el valor de todas las partículas del set $\widehat{S}_{t-1} = \left\{ \widehat{x}_{t-1}^{(i)}, 1/n \right\}_{i=1}^n \cong p(\widehat{x}_{t-1}^{(1:n)} | \widehat{y}_{1:t-1})$ y generar la salida $S_{t|t-1} = \left\{ \widehat{x}_{t|t-1}^{(i)}, 1/n \right\}_{i=1}^n \cong p(\widehat{x}_t^{(1:n)} | \widehat{y}_{1:t-1})$, con independencia del sistema al que cada una de ellas esté asociada en la estimación multimodal desarrollada (ver la Figura 6.1). La generalidad de este modelo aporta la flexibilidad que requiere el tracking de distintos tipos de objetos, y además facilita el proceso de asociación, imprescindible en este tipo de aplicaciones.

c) Modificaciones en el paso de corrección

En el paso de corrección de la versión básica del PF, el algoritmo “*Bootstrap*”, se calcula el peso de las partículas $\widetilde{w}_t = \left[\widetilde{w}_t^{(i)} \right]_{i=1}^n$ a través del modelo de observación $p(\widehat{y}_t | \widehat{x}_t) \cong h(\widehat{x}_{t|t-1}^{(1:n)}, \widehat{o}_t)$ del sistema bajo estudio. Por otro lado, en la aplicación del filtro al problema de estimación multimodal de interés, en esta etapa se incluye un proceso de asociación que especifica la relación concreta entre cada partícula $\widehat{x}_t^{(i)} / i = 1:n$ (o cada modo de la creencia) y cada medida del conjunto $Y_t = \left\{ \widehat{y}_{j,t} \right\}_{j=1}^{m_t}$ extraído del entorno en ese instante t .

Con este objetivo, el cálculo de los pesos sin normalizar de las partículas $\widetilde{w}_t = \left[\widetilde{w}_t^{(i)} \right]_{i=1}^n$ en el XPF se realiza mediante la expresión <4.45> del capítulo 4, que se repite a continuación:

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot p(\widehat{y}_t | \widehat{x}_t^{(i)}) = w_{t-1}^{(i)} \cdot e^{-d_{\min,i,t}^2 / 2\sigma} / i = 1:n, \quad <6.7>$$

donde $d_{\min,i,t}$ es una medida de relación, y más concretamente distancia, entre el valor de la partícula $\widehat{x}_t^{(i)}$ y el del dato del set Y_t más próximo en el espacio de definición del modelo de observación (espacio de posición XYZ en la aplicación de MTT de interés), que se supone la asociada a $\widehat{x}_t^{(i)}$, según el método de asociación NN empleado en este estimador.

$$d_{\min,i,t} = \min_{j=1:m_t} \left\{ d \left(h(\widehat{x}_{t|t-1}^{(i)}), \widehat{y}_t^{(j)} \right) \right\} / i = 1:n \quad <6.8>$$

Sin embargo, como se expone en el capítulo 4, este procedimiento es poco robusto, pues asigna pesos mayores a aquellas partículas que representan a hipótesis con un modelo de observación más fiable, y provoca la baja ponderación, y la probable eliminación del set S_t en el paso de selección posterior, de las que representen a hipótesis más pobremente o ruidosamente sensadas. Esta falta de robustez imposibilita el desarrollo del proceso de corrección con la fiabilidad necesaria para que el set de salida $S_t' = \left\{ \widehat{x}_{t|t-1}^{(i)}, \widetilde{w}_t^{(i)} \right\}_{i=1}^n$ represente el estado de los múltiples objetos implicados en la tarea de MTT.

Para mejorar la robustez de este paso, en el XPFCP se incluyen también varias modificaciones a su funcionamiento:

1. Se emplea la propuesta de clasificación determinística presentada en el capítulo 5 para organizar de forma robusta (con los valores de fiabilidad antes expuestos) el conjunto de medidas en clases $Y_t \Rightarrow G_{j,t|in} / j=1:k_{in,t}$. De este modo se obtiene un representante de cada grupo, el centroide $\bar{g}_{j,t|in}$, que resulta del filtrado de cada grupo de medidas $L_{j,t|in}$, y presenta, por tanto, una exactitud equivalente y alta para todos los conjuntos, independientemente de la densidad y fiabilidad de las medidas incluidas en cada grupo. El conjunto de clases generado por el clasificador en este punto $Y_t \Rightarrow G_{1:k,t|in}$ es el usado, además, en el paso de re-inicialización de la ejecución siguiente (en $t + 1$) del XPFCP.
2. Se utiliza este representante robusto de cada clase de medidas $\bar{g}_{j,t|in} / j=1:k_{in}$ en el proceso de asociación implícito en el pesado, suponiendo, como se comenta en capítulos anteriores y se justifica con los datos de fiabilidad de los clasificadores, que cada grupo de medidas $L_{j,t|in}$ se obtiene del modelo de observación de cada una de las hipótesis implicadas en el proceso de estimación multimodal. Con este nuevo método de asociación, el proceso de cálculo de los pesos $w_t^{(i)}$ se modifica en el XPFCP, respecto al del XPF definido en las expresiones <6.7> y <6.8>, quedando como sigue:

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot p(\bar{g}_{1:k,t|in} | \bar{x}_t^{(i)}) = w_{t-1}^{(i)} \cdot e^{-d_{\min,i,t}^2 / 2\sigma} / i = 1:n, \quad <6.9>$$

donde la distancia $d_{\min,i,t}$ en este caso establece la relación entre el valor de la partícula $\bar{x}_t^{(i)} / i=1:n$ y el del centroide $\bar{g}_{j,t|in} / j=1:k_{in}$ más próximo a ésta en el espacio de definición del modelo de observación XYZ , como muestra la ecuación siguiente:

$$d_{\min,i,t} = \min_{j=1:k} \{d(h(\bar{x}_{t-1}^{(i)}), \bar{g}_{j,t|in})\} / i = 1:n \quad <6.10>$$

Se incluye así en el XPFCP una asociación de tipo NN, entre el conjunto de medidas organizado en clases y el de hipótesis de seguimiento (en la aplicación de MTT de interés). Esta asociación es funcionalmente igual a la del XPF, pero de mayor robustez y fiabilidad que ésta, al basarse en los centroides de las clases $\bar{g}_{j,t|in}$, y gracias a la asociación (entre las medidas de cada grupo $G_{j,t|in} / j=1:k_{in}$ y los hipotéticos objetos de las que se extraen) que desarrollan de forma implícita los clasificadores descritos en el capítulo 5.

3. Finalmente, el peso $w_t^{(i)}$ de cada partícula $\bar{x}_t^{(i)} / i=1:n$ se pondera por la verosimilitud de validación $\frac{P_{j,t|in}}{P_{valid}}$ de la clase de medidas con cuyo centroide $\bar{g}_{j,t|in}$ se asocia la partícula a través de la distancia $d_{\min,i,t}$, valor de verosimilitud que pasa a llamarse $p_{\min,i,t}$. La verosimilitud de validación de cada clase se define, por tanto, como el número de veces

que la verosimilitud de la clase $p_{j,t|in}$ supera el límite p_{valid} necesario para que la clase sea validada por probabilidad (ver apartado 5.2.5 del capítulo 5). El objetivo de esta ponderación es equilibrar la probabilidad de las hipótesis de seguimiento representadas por el set de partículas a la salida del paso de corrección $S'_t = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$, en función de la fiabilidad de la clase de medidas con la que se asocia cada una de ellas. Como se demuestra en un apartado posterior, este procedimiento robustece la representación de las hipótesis en la creencia de salida del XPFCP. Con esta última modificación, la ecuación empleada para el cálculo de los pesos $w_t^{(i)} / i = 1:n$ mencionado, a falta de la normalización (ver ecuación <4.46> en el capítulo 4), queda como sigue:

$$w_t^{(i)} = p_{\min,i,t} \cdot w_{t-1}^{(i)} \cdot e^{-d_{\min,i,t}^2 / 2\sigma} / i = 1:n, \quad <6.11>$$

donde la función de distancia $d_{\min,i,t}$ mantiene la forma especificada por la ecuación <6.10>. La ponderación comentada implica además la incorporación en la creencia generada por el XPFCP $p(\bar{x}_t^{n-n_{m,t}} | \bar{y}_{1:t})$ de más información sobre las clases de medidas $Y_t \Rightarrow G_{1:k,t|in}$, es decir, sobre la verosimilitud de cada hipótesis implicada en la tarea de MTT de interés.

Obtenido el valor de cada peso $w_t^{(i)} / i = 1:n$, este se normaliza, del mismo modo que se hace en el PF básico, a través de la ecuación <4.18> del capítulo 4 que se repite aquí:

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^n w_t^{(i)}} \quad <6.12>$$

De los análisis expuestos en párrafos anteriores se extrae que, gracias al uso de las clases de medidas $Y_t \Rightarrow G_{1:k,t|in}$ en la etapa de corrección del XPFCP, el proceso de asociación incluido en esta etapa de ponderación está formado por dos partes:

- Una primera que es intrínseca al proceso de clasificación de medidas $Y_t \Rightarrow G_{1:k,t|in}$, como se expone en el capítulo 5. Se trata de una asociación de tipo NN que establece el proceso de identificación de clases al evolucionar bajo la suposición de que el conjunto de medidas $L_{j,t|in}$ que cada una de ellas agrupan ($j = 1:k_{in,t}$) proceden del modelo de observación de cada target concreto en la aplicación de MTT de interés. De las dos propuestas de clasificación presentadas, el “*k-medias secuencial*” con validación muestra mayor fiabilidad en esta tarea de identificación, y por ende, de asociación, por lo que, tal y como se justifica teóricamente en dicho capítulo y empíricamente en apartados posteriores de éste, este algoritmo se plantea como solución más adecuada para llevar a cabo la tarea de clasificación de medidas incluida en la funcionalidad del XPFCP.

- Una segunda que se lleva a cabo en la etapa de corrección del XPFCP, concretamente en el proceso de búsqueda del centroide $\bar{g}_{j,t|in}$ más próximo, en el espacio de observación, a cada hipótesis de estimación representada por el valor predicho de la partícula correspondiente $\bar{x}_{t|t-1}^{(i)}$ a la entrada de esta etapa del filtro. Esta segunda fase de asociación es también de tipo NN y, como se explica en el capítulo 2, no tiene por sí sola la fiabilidad que requiere la aplicación de MTT de interés. La primera parte del proceso de asociación aporta a este algoritmo de asociación la fiabilidad y robustez que no tiene y que desaconseja su uso en aplicaciones como la de interés frente al de métodos más fiables como el JPDA, que tienen, en contrapartida, una alta carga computacional. La robustez y fiabilidad de la solución generada por el clasificador aseguran el correcto funcionamiento de este sistema de asociación en dos fases basado en el algoritmo NN, y su uso facilita la ejecución en tiempo real de la estimación multimodal.

La comentada fiabilidad del proceso de asociación así descrito queda demostrada con los resultados presentados más adelante en este capítulo. Por otra parte, en el capítulo siguiente se comparan estos resultados con los generados por otras propuestas de tracking, más extendidas entre la comunidad científica, y basadas en algoritmos de asociación de tipo JPDA.

El uso de los valores de salida del clasificador de medidas $Y_t \Rightarrow G_{1:k,t|in}$ (el centroide $\bar{g}_{1:k,t|in}$ y la verosimilitud de las clases $p_{1:k,t|in}$) en la función de pesado $w(\bar{x}_{0:t}) \rightarrow \bar{w}_t = [\tilde{w}_t^{(i)}]_{i=1}^n$, y la creación del proceso de asociación de tipo NN en este paso de corrección a partir de ellos, conlleva que la PDF $p(\bar{g}_{1:k,t|in} | \bar{x}_t^{(i)})$ base de esta función, presente una semejanza a la creencia a priori $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t-1})$, representada por el set $S_{t|t-1} = \{\bar{x}_{t|t-1}^{(i)}, 1/n\}_{i=1}^n$, más robusta que la usada en el XPF ($p(\bar{y}_t | \bar{x}_t^{(i)}) = norm(h(\bar{x}_{t|t-1}^{(i)}), O_t)$), y en el algoritmo “*Bootstrap*” ($p(\bar{y}_t | \bar{x}_t)$).

La robustez de la relación entre el valor predicho de las hipótesis $\bar{x}_{t|t-1}^{(i)}$ de estimación y la PDF de verosimilitud elegida $p(\bar{g}_{1:k,t|in} | \bar{x}_t^{(i)})$, hacen que la función $w(\bar{x}_{0:t}) \rightarrow \bar{w}_t = [\tilde{w}_t^{(i)}]_{i=1}^n$ propuesta para el paso de corrección del XPFCP en la ecuación <6.11> genere valores de pesado más altos para todas las hipótesis representadas por el set $S_{t|t-1}$ que los que genera el XPF o el algoritmo “*Bootstrap*”, independientemente de la fiabilidad del modelo de observación con el que se asocia cada una.

Este análisis permite concluir que la solución expuesta para el paso de corrección del XPFCP cumple el objetivo planteado al principio de este punto: eliminar el problema de empobrecimiento, característico de los algoritmos usados como base del XPFCP (el XPF y el “*Bootstrap*”), mantenido el bajo coste computacional que los hace más adecuados para aplicaciones con requerimientos de tiempo real como la de interés, que otros algoritmos planteados por la comunidad científica (p. ej. el APF o el EKPF analizados en el capítulo 4).

Más adelante se justifica mediante resultados empíricos que el set a la salida del paso de corrección del XPFCP $S_t' = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ no está empobrecido, usando para ello como factor de calidad el número aproximado de partículas eficaces \hat{n}_{eff} , descrito en el capítulo 4.

La Figura 6.2 demuestra los análisis expuestos en los párrafos anteriores, a través de un ejemplo de funcionamiento del XPFCP. En ella se observa que el peso asociado a las partículas insertadas en el paso de re-inicialización (representadas como rombos de color verde) $w_t^{(i)} / i = n - n_{m,t-1} : n$ es suficientemente elevado (en relación al del resto $w_t^{(i)} / i = 1 : n - n_{m,t-1}$) como para que estas se reproduzcan en el paso de selección que sigue al de corrección, efecto que no se consigue en el XPF (ver la Figura 4.6 en el capítulo 4).

Más concretamente, las 2 partículas con valor intermedio de x en la creencia $\hat{p}(x_{t-1}^{(1:n)} | y_{1:t-1})$ a la salida del paso de re-inicialización en la Figura 6.2, que representan a una hipótesis de estimación inexistente en la creencia a la entrada de este paso $p(x_{t-1}^{(1:n-n_{m,t-1})} | y_{1:t-1})$, adquieren un peso en el paso de corrección que conlleva su reproducción a la salida del de selección.

De esta forma, el set final $S_t = \{x_t^{(i)}, 1/n\}_{i=1}^{n-n_{m,t}}$ a la salida de cada ejecución del XPFCP está distribuido de forma equilibrada entre todas las hipótesis estimadas por el filtro multimodal (alrededor de 3 partículas asociadas con cada hipótesis en el ejemplo mostrado en la Figura 6.2), independientemente de que las hipótesis estuviesen o no representadas en el set de entrada al XPFCP en dicha ejecución $S_{t-1} = \{x_{t-1}^{(i)}, 1/n\}_{i=1}^{n-n_{m,t-1}}$.

Queda así demostrado que el uso de la salida del clasificador determinístico $Y_t \Rightarrow G_{1:k,t|in}$ en la etapa de corrección del XPFCP refuerza el peso $\tilde{w}_t^{(i)}$ de las partículas insertadas en el paso de re-inicialización $\hat{s}_{i,t-1} / i = n - n_{m,t-1} : n$, asignándoles una alta probabilidad de que se reproduzcan en el posterior paso de selección, desde el instante t en el que se incorporan al set S_t ; y permitiéndoles por tanto representar en la creencia multimodal de salida del XPFCP $p(\bar{x}_t | \bar{y}_{1:t})$ desde ese primer instante t , a las hipótesis de estimación con las que se asocian. Además, queda así también demostrado desde un punto de vista teórico (y corroborado de forma empírica en un apartado posterior), que este objetivo se alcanza con un número pequeño de partículas insertadas en el paso de re-inicialización $n_{m,t-1}$, de modo que se mantiene el rigor matemático de la representación discreta de esta creencia.

d) Modificaciones en el paso de selección

El paso de selección del XPFCP es funcionalmente idéntico al del XPF: las partículas del set a su entrada $S_t' = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ se reproducen en función de su peso, quedando todas ellas con un peso idéntico, y generándose además a la salida del paso el número $n - n_{m,t}$ de éstas

necesario para que pueda ejecutarse la etapa de re-inicialización en la ejecución siguiente del filtro. El paso de selección es, por ello, necesario en el flujograma del XPFCP y el conjunto de partículas a la salida del paso de selección conforman la representación discreta final de la creencia multimodal generada por el filtro bayesiano $p(x_t^{(1:n-n_{m,t})} | y_{1:t})$.

Tal y como se explica al principio de este apartado, el valor en que se debe disminuir el número de partículas a la salida del XPFCP $n_{m,t}$ depende del conjunto de clases obtenido de la segmentación del set de medidas $Y_t \Rightarrow G_{1:k,t|in}$ en este instante de tiempo t . Más concretamente, y como se especifica en detalle en el apartado siguiente, este valor $n_{m,t}$ depende del número de clases nuevas generadas y validadas por el clasificador de medidas.

Como se comenta en el capítulo 4, la elección de uno u otro algoritmo de entre los propuestos por la comunidad científica para desarrollar la tarea de selección descrita, no es determinante en la eficiencia, fiabilidad o robustez de la estimación obtenida. En la aplicación de estimación multimodal a la tarea de MTT de interés en la tesis, se ha empleado un algoritmo de selección residual modificado en los siguientes puntos respecto a su versión básica (propuesta en [Liu98] y descrita en el apartado 4.1.3 del capítulo 4):

1. *La primera etapa del algoritmo de resampling se modifica respecto a la del algoritmo descrito en [Liu98]:* como en el algoritmo básico, se obtiene el número de “hijos” a extraer de cada partícula $\bar{x}_{t|t-1}^{(i)} / i = 1:n$ a través del valor normalizado de su peso $\tilde{w}_t^{(i)}$, pero ponderándolo en este caso por el número de partículas $n_{m,t}$ que se desea obtener a la salida de este paso del XPFCP, como se muestra en la siguiente expresión:

$$n_i = (n - n_{m,t}) \cdot \tilde{w}_t^{(i)} / i = 1:n \quad <6.13>$$

2. *La segunda etapa del algoritmo es idéntica a la del algoritmo descrito en el capítulo 4:* se reparten el número de muestras que falte por asignar (en el XPFCP $(n - n_{m,t}) - \sum_{i=1}^n n_i$) de forma aleatoria entre todo el set de partículas de entrada $S'_t = \{\bar{x}_{t|t-1}^{(i)} / i = 1:n\}$, mediante el procedimiento de selección multinomial.

De este modo, el algoritmo de resampling residual propuesto en [Liu98] y de uso muy extendido en la aplicación en tiempo real del PF debido a su bajo coste computacional, se adapta a la funcionalidad impuesta para la tarea de re-muestreo en el XPFCP.

Al margen de lo expuesto en los párrafos anteriores, en un apartado posterior se analiza el efecto que el uso de uno u otro algoritmo de resampling tiene en la fiabilidad y robustez del XPFCP, demostrándose mediante el estudio de los resultados generados por el filtro con distintos algoritmos de selección, la mínima influencia de la elección en estos parámetros.

6.1.2. Otras consideraciones sobre el XPFCP

Analizada, de forma general, la funcionalidad específica de cada paso del XPFCP, así como los aspectos en los que ésta aporta una mejora a la de otros estimadores bayesianos basados en

el PF, en este apartado se describen de forma detallada ciertas características de la funcionalidad de alguna de estas etapas que quedan fuera del análisis general expuesto en el apartado 6.1.1.

a) En el paso de re-inicialización. Definición del parámetro $n_{m|j,t-1}$ en función de la novedad de la clase de medidas $G_{j,t-1|in} / j=1:k_{in,t-1}$

Como se comenta en el apartado 6.1.1, al calcular el número de partículas que se añaden asociadas con cada clase de medidas $n_{m|j,t-1} / j=1:k_{in,t-1}$ se tienen en cuenta dos factores:

1. *El número de clases generadas de Y_{t-1} , $k_{in,t-1}$.*
2. *La novedad de cada clase $j=1:k_{in,t-1}$ en el conjunto $G_{j,t-1|in}$, caracterizada por el parámetro booleano $\alpha_{init,j,t-1}$, que toma valor 1 si $t-1$ es el primer instante en el que aparece la clase, y el valor 0 si la clase no es nueva en ese instante.*

A partir de los dos factores, la expresión usada para calcular el número de partículas por clase de medidas $n_{m|j,t-1} / j=1:k_{in,t-1}$ a insertar en el paso de re-inicialización, ya mostrada de forma genérica en la ecuación <6.5>, queda como sigue:

$$n_{m|j,t-1} = g(k_{in,t-1}, G_{j,t-1|in}) = \frac{1}{k_{in,t-1}} \cdot (n_m + n_{init} \cdot \alpha_{init,j,t-1}) / j=1:k_{in,t-1}, \quad <6.14>$$

donde n_m es el valor mínimo de partículas que se añaden en este paso de re-inicialización y coincide con el constante usado en el XPF (ver, por ejemplo ecuación <6.1>), y n_{init} es el número de partículas adicional que se añade en este paso para potenciar la representación de cada nueva hipótesis de estimación en la creencia $\hat{p}(\bar{x}_{t-1} | \bar{y}_{1:t-1})$.

De este modo, el número total de partículas $n_{m,t-1}$ insertadas en el set de entrada al paso de re-inicialización $S_{t-1} \cong p(\bar{x}_{t-1}^{(1:n-n_{m,t-1})} | \bar{y}_{1:t-1})$ para dar lugar al set de salida $\hat{S}_{t-1} \cong \hat{p}(\bar{x}_{t-1}^{(1:n)} | \bar{y}_{1:t-1})$ se obtiene al unificar las expresiones <6.5> y <6.14>:

$$n_{m,t-1} = \sum_{j=1}^{k_{in,t-1}} n_{m|j,t-1} = \sum_{j=1}^{k_{in,t-1}} \frac{1}{k_{in,t-1}} \cdot (n_m + n_{init} \cdot \alpha_{init,j,t-1}) \quad <6.15>$$

Según la expresión <6.15>, el valor de $n_{m,t}$ varía en cada instante temporal t , por tanto, entre un mínimo de n_m (usado en caso de que el clasificador no genere ninguna clase de medidas nueva en $G_{1:k,t|in}$) y un máximo de $n_m + k_{in,t-1} \cdot n_{init}$ (usado si todas las clases $G_{1:k,t-1|in}$ generadas por el clasificador son nuevas en t).

El valor de $n_{m,t}$ se calcula, por tanto, a partir de la expresión <6.15> en el paso de selección del XPF en cada instante t , pues define el número de partículas que se generan a su salida

$n - n_{m,t}$, y que conforman la representación discreta de la salida global de estimación multimodal del filtro, $S_t \cong p(\bar{x}_t^{(1:n-n_{m,t})} | \bar{y}_{1:t})$ en dicho instante temporal t . El paso de selección es por ello imprescindible para el desarrollo del proceso de inserción de $n_{m,t}$ partículas realizado en el paso de re-inicialización (como ocurre también en el XPF) si se desea que el número total de partículas n del XPFCP no se incremente con el tiempo.

El ajuste del nuevo parámetro n_{init} se analiza, junto al del resto de parámetros del XPFCP, en un apartado posterior. Allí se comprueba que el incremento del valor de los parámetros n_m y n_{init} , y por ende de $\gamma_t = \frac{n}{n_{m,t-1}}$, no es sinónimo de un incremento de la fiabilidad y robustez del XPFCP, como se adelanta en párrafos anteriores. Además es necesario remarcar que, debido a su significado (ver expresión <6.15>), el valor del parámetro n_{init} ha de ser limitado de modo que $n_{m,t-1}$ sea menor al número total de partículas n , para que el set de partículas de salida del paso de corrección $S_t = \left\{ \bar{x}_t^{(i)}, \frac{1}{n - n_{m,t}} \right\}_{i=1}^{n-n_{m,t}}$ no quede vacío en situaciones complejas de estimación (con un alto número de hipótesis, y por ende de clases de medidas $k_{in,t-1}$). Para considerar esta premisa, en los análisis incluidos más adelante se estudia en primer lugar el efecto que tiene la modificación de $n_{m,t-1}$ en la eficiencia del XPFCP, suponiéndolo constante, para después abordar la sensibilidad del filtro frente al valor de n_{init} .

Por otro lado, se recuerda en este punto que las dos propuestas de segmentación presentadas en el capítulo 5 incluyen un proceso de validación de clases de velocidad modulable a través de distintos parámetros, siendo el más directo β_{valid} . Tal y como se analiza más adelante, a través de este parámetro es posible modular, por tanto, la velocidad con la que las distintas hipótesis de seguimiento se incorporan a la distribución multimodal discreta estimada por el XPFCP $p(\bar{x}_t^{(1:n-n_{m,t})} | \bar{y}_{1:t})$, y por ende a la tarea de MTT de interés, según los objetos son detectados en el entorno y la matriz Y_t incorpora medidas relacionadas con ellos. Este procedimiento es equivalente al de persistencia empleado en los procesos de “*birth*” y “*death*” que incluye, por ejemplo, el algoritmo de MTT expuesto en [Ng06] y comentado en el capítulo 2.

Finalmente, conviene destacar que la obtención del parámetro $\alpha_{init,j,t}$ para cada clase de medidas $j=1:k_{in,t}$ es posible, gracias al sistema de identificación implícito también en las propuestas de clasificación. Para ello es necesario comparar los identificadores de los grupos en la salida generada por el clasificador en el instante $t-1$, $G_{1:k,t-1|in} \Rightarrow \tau_{1:k_{in,t-1}}$, con los de su salida generada en t , $G_{1:k,t|in} \Rightarrow \tau_{1:k_{in,t}}$. A las clases $j=1:k_{in,t}$ del conjunto $G_{1:k,t|in}$ cuyo identificador no se encuentre en el listado obtenido de $G_{1:k,t-1|in} \Rightarrow \tau_{1:k_{in,t-1}}$ se les asigna un parámetro $\alpha_{init,j,t} = 1$, y al resto un parámetro $\alpha_{init,j,t}$ nulo.

b) En el paso de corrección. Definición de la función de verosimilitud $p(\bar{g}_{1:k,t|in} | \bar{x}_t^{(i)})$

La forma de la verosimilitud $p(\bar{g}_{1:k,t|in} | \bar{x}_t^{(i)})$ que define el peso de cada partícula $\tilde{w}_t^{(i)} / i = 1:n$ en el XPFCP a través de la expresión <6.11>, es igual a la usada en el XPF con este mismo objetivo ($p(\bar{y}_t | \bar{x}_t^{(i)})$ mostrada en la ecuación <6.7>): una gaussiana definida para cada partícula $\bar{x}_{t|t-1}^{(i)}$ en el espacio de distancias, cuya media viene dada por la proyección del valor de la partícula al espacio de medidas ($h(\bar{x}_{t|t-1}^{(i)})$), y cuya matriz de covarianza O queda definida por el modelo del ruido de medida \bar{o}_t establecido para la aplicación (ver la Figura 4.7 del capítulo 4). Es decir, la funcionalidad del paso de corrección del XPFCP, al igual que la del XPF, se basa en la gaussiana $p(\bar{y}_t | \bar{x}_t^{(i)}) = norm(h(\bar{x}_{t|t-1}^{(i)}), O)$.

La diferencia entre los dos algoritmos en este paso está en las realizaciones de la gaussiana $norm(h(\bar{x}_{t|t-1}^{(i)}), O)$ entre las que se busca el valor que define el peso $\tilde{w}_t^{(i)} / i = 1:n$ de la partícula correspondiente: en el XPFCP se usan los centroides $\bar{g}_{1:k,t|in}$ de las clases obtenidas del set de medidas $Y_t \Rightarrow G_{1:k,t|in}$, mientras que en el XPF se usa el propio set $Y_t = \{\bar{y}_t^{(j)}\}_{j=1}^{m_t}$.

La Figura 6.3 es una representación unidimensional del proceso descrito, equivalente a la ya mencionada Figura 4.7 del capítulo 4. La comparación de estas dos figuras facilita la comprensión de las diferencias descritas en el párrafo anterior, y permite observar la mayor robustez de la propuesta de corrección planteada para el XPFCP frente a la del XPF.

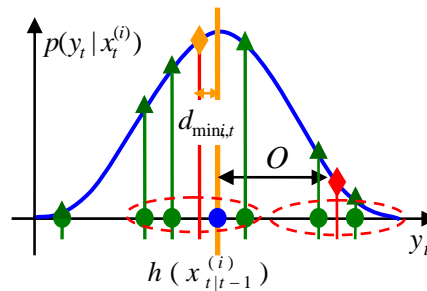


Figura 6.3. Representación unidimensional del proceso de obtención de los pesos en el paso de corrección del XPFCP. La línea azul muestra la gaussiana de verosimilitud usada $p(g_t | x_t^{(i)})$. El valor de distancia base del cálculo de $\tilde{w}_t^{(i)} / i = 1:n$ se calcula entre la media $h(x_{t|t-1}^{(i)})$ de la gaussiana (con línea naranja) y el centroide más próximo $g_{1:k,t|in}$ (en naranja igual que la distancia). En rojo se muestran las agrupaciones de medidas $Y_t \Rightarrow G_{1:k,t|in}$ (representadas con puntos verdes), cuyo centroide se indica con un rombo del mismo color.

Del análisis expuesto se concluye además, que la propuesta de cálculo de los pesos $\bar{w}_t = [\tilde{w}_t^{(i)}]_{i=1}^n$ diseñada para el XPFCP tiene una menor carga computacional que la planteada para el XPF, por lo que facilita la consecución de la especificación de tiempo real: el proceso de

segmentación $Y_t \Rightarrow G_{1:k,t|in}$, comprime la información contenida en el conjunto de m_t medidas en $k_{t,in}$ clases, con lo que se reduce a un número mucho menor de posibilidades la búsqueda de la distancia mínima $d_{\min,i,t}$, necesaria para calcular $\bar{w}_t = \left[\tilde{w}_t^{(i)} \right]_{i=1}^n$, según la ecuación <6.11>.

Por otra parte, esta función de distancia $d_{\min,i,t} / i = 1:n$, determinada de forma general por la ecuación <6.10>, se especifica en la aplicación de interés como una distancia euclídea ($d_{euclídea|i,j} / j = 1:k_{in,t}$), con lo que su definición se reescribe como sigue:

$$d_{\min,i,t} = \min_{j=1:k} \left\{ \sqrt{\left(h(\bar{x}_{t|t-1}^{(i)}) - \bar{g}_{j,t|in} \right)^2} \right\} / i = 1:n \quad <6.16>$$

La razón fundamental por la que se elige esta forma para la medida de distancia en el XPFCP es la misma que determina su aplicación en las propuestas de clasificación descritas en el capítulo 5 (ver el apéndice II): las características del conjunto de datos sobre el que se desarrolla la aplicación de MTT de interés (establecidas en el capítulo 1, fundamentalmente su procedencia de un único sensor), así como la menor complejidad de cómputo de esta forma respecto a otras con similares características de invarianza (como la de Mahalanobis), determinan esta elección.

A partir de la especificación de la medida de distancia se puede además simplificar la matriz de covarianza O , característica del modelo de observación y usada en el cálculo de los pesos como una matriz diagonal cuyos elementos no nulos están definidos por la varianza $\sigma_{o,i}^2$ de las medidas de posición en las tres dimensiones del espacio de observación propuesto ($i = \{x, y, z\}$), tal y como se comenta ya en el capítulo 2. La forma de esta matriz queda como se muestra:

$$O = \begin{bmatrix} \sigma_{o,x}^2 & 0 & 0 \\ 0 & \sigma_{o,y}^2 & 0 \\ 0 & 0 & \sigma_{o,z}^2 \end{bmatrix} \quad <6.17>$$

La elección del valor específico de la matriz de covarianza O (los valores de varianza $\sigma_{o,x}^2$, $\sigma_{o,y}^2$, $\sigma_{o,z}^2$) que se emplea en los experimentos mostrados, así como el efecto de la variación de este parámetro en la fiabilidad del XPFCP se estudia más adelante en este capítulo.

Se observa así que la etapa de corrección propuesta para el XPFCP, al igual que la del XPF, cumple el objetivo de fortalecer el conjunto de pesos $\bar{w}_t = \left[\tilde{w}_t^{(i)} \right]_{i=1}^n$ (para disminuir el problema de empobrecimiento ya comentado), desencadenante de propuestas de estimación más complejas como el EKPF y el APF (analizadas en el capítulo 4). El algoritmo de corrección aquí expuesto mejora además el comportamiento del XPF en este punto, pues elimina el problema de empobrecimiento que persiste en el XPF, y reduce la carga computacional del filtro, como demuestran los resultados empíricos expuestos más adelante.

6.1.3. Sensibilidad y ajuste de los parámetros del XPFCP

En este punto se incluye el análisis del efecto que tienen las mejoras y parámetros del XPFCP descritos en secciones anteriores, en su aplicación a la tarea de MTT de interés. En las gráficas aquí incluidas se muestra la mejora que experimentan los resultados obtenidos con el XPFCP conforme los parámetros analizados son ajustados, remarcándose además en cada caso la sensibilidad del algoritmo al valor de cada parámetro.

Para analizar la eficiencia del estimador multimodal y confirmar que la propuesta presentada en la tesis cumple con las especificaciones establecidas en el capítulo 1 en la aplicación de interés, se emplean las medidas de tasa de error, sensibilidad y t_{exe} usadas en las comparaciones empíricas del capítulo 5. Además, para facilitar la comprensión del análisis, estos datos se extraen de las mismas pruebas referidas en el mismo capítulo (descritas en el apéndice III).

Al final de este capítulo se muestran, a modo de conclusión de la tesis, resultados de la aplicación del XPFCP a un experimento de seguimiento de tipología distinta a la de los usados en los estudios incluidos a continuación y en el capítulo 5. De este modo, se facilita por un lado el análisis comparativo de las distintas técnicas expuestas en la tesis, y se demuestra por otro la flexibilidad de aplicación de la propuesta de estimación multimodal.

Es necesario destacar que los datos sobre tasa de error expuestos en este y en los siguientes apartados se refieren a la no asignación de partículas a una hipótesis de seguimiento. Teniendo en cuenta que, al ser muestras discretas de la creencia a la salida del estimador bayesiano

$p(\bar{x}_t | \bar{y}_{1:t})$, cada partícula del set $S_t = \left\{ \bar{x}_t^{(i)}, \frac{1}{n - n_{m,t}} \right\}_{i=1}^{n - n_{m,t}}$ no representa a una hipótesis de

seguimiento distinta $j = 1:k_{out,t}$ sino que el conjunto S_t se distribuye alrededor de su centro de gravedad, más o menos próximo a éste en función del set de medidas de entrada al filtro Y_t y de la dispersión del set S_t . No se puede hablar, por tanto, de tasas de error por duplicación o desplazamiento de alguna de las hipótesis o por la unión de 2 hipótesis en 1, como se hace en el capítulo 5. La referencia a estos tipos de errores vuelve a usarse en análisis, incluido en el apartado 6.2, sobre los resultados generados por el clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$, en los que sí debe haber una sola hipótesis de tracking (posición, identidad y velocidad) por cada uno de los $k_{out,t}$ objetos seguidos, centrada en su centro de gravedad.

Además de las medidas de tasa de error comentadas, en la exposición aquí incluida se emplean otros parámetros más específicos del análisis de fiabilidad de los estimadores basados en el PF, como el número aproximado de partículas eficaces \hat{n}_{eff} o el número total de éstas n necesario para asegurar su fiabilidad. El valor de \hat{n}_{eff} generado por el algoritmo es un indicador de la eficiencia del estimador multimodal propuesto frente a distintos factores del proceso, como la inexactitud del modelo de estimación (y su robustez y flexibilidad frente a cambios en éste) o la disminución del número de partículas usadas (y por ende su t_{exe}). Finalmente, como

se expone en [Gustafsson02], la eliminación de los problemas de empobrecimiento y degeneración propios del PF puede asegurarse con la obtención de valores de $\hat{n}_{eff} > \frac{2}{3} \cdot n$.

a) Efecto de la modificación de n

La Figura 6.4 muestra la tasa de error obtenida con el XPFCP propuesto, ejecutado en dos experimentos distintos (en azul los resultados obtenidos en el test complejo, en verde los obtenidos en el completo corto y en rojo la media de ambos), al modificar el número total de partículas del algoritmo n de 100 a 1000, con incrementos de 100 unidades.

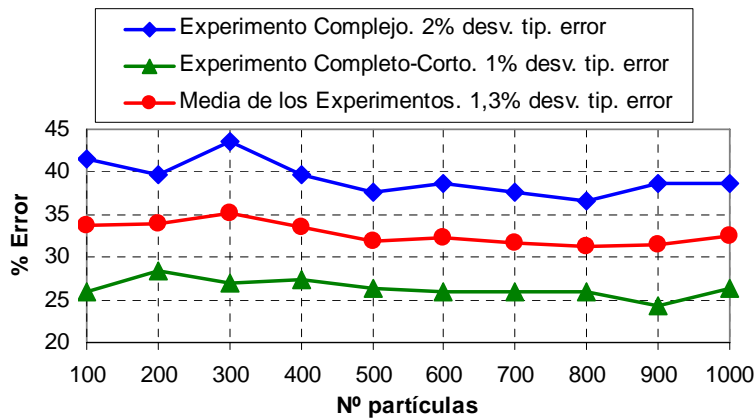


Figura 6.4. Tasa de error generada por el XPFCP en función del número de partículas n de éste, al ejecutarlo en el experimento complejo (en azul) y en el completo corto (en verde), así como la media de ambos resultados (en rojo).

Como se puede observar en la Figura 6.4, la variación de la tasa de error es prácticamente nula en los dos experimentos, lo que demuestra que, aún bajo las complejas situaciones de testeo, el incremento de n no aporta ninguna mejora en la fiabilidad de la estimación. Esta conclusión es especialmente importante pues resuelve una de las dudas que se plantean en la literatura del área y que se esgrimen como inconveniente de uso del PF en tareas de estimación: *¿Cuántas partículas son necesarias en cada aplicación?*

La pregunta anterior surge entre la comunidad científica, fundamentalmente debido al efecto que el aumento de n tiene en el t_{exe} del estimador bayesiano. En efecto, como muestra la Figura 6.5, aparece una diferencia de casi el 50% en el t_{exe} ¹ del algoritmo, ejecutado en el test complejo, al modificar el valor de n aproximadamente en la misma magnitud (gráfica en azul oscuro con el t_{exe} para $n = 600$, en azul claro para $n = 300$ y en azul intenso para $n = 900$).

Las gráficas izquierda y derecha de la Figura 6.6 muestran, respectivamente, el valor medio de este t_{exe} , y la cantidad de éste empleado por partícula, en función del número de partículas n usado en el filtro bayesiano, en las mismas condiciones que los resultados mostrados en la Figura 6.4. La Figura 6.6 permite llegar a la misma conclusión expuesta en el párrafo anterior: el valor medio del t_{exe} del algoritmo crece linealmente con n (gráfica izquierda) y de forma

¹ Las medidas de t_{exe} mostradas en este capítulo y en el siguiente se realizan sobre una plataforma de ejecución en tiempo real desarrollada C, y corriendo en un procesador Intel Centrino DUO a 1.8GHz.

independiente a la complejidad de la tarea de estimación (a partir de un valor mínimo de partículas, como muestra la grafica derecha), que en la aplicación de interés se refiere fundamentalmente al número de hipótesis de seguimiento distintas representadas en la creencia multimodal y discreta $p(\bar{x}_t^{n-n_{m,t}} | \bar{y}_{1:t})$ a la que caracteriza el set de partículas S_t .

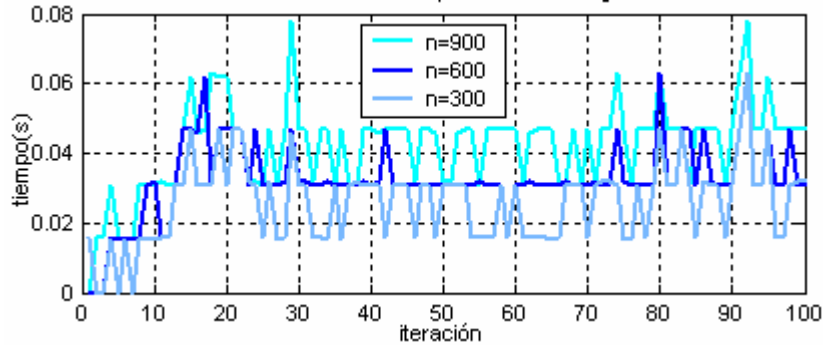


Figura 6.5. Gráficas comparativas del t_{exe} del XPFCP en función de n , obtenidas en el experimento complejo.

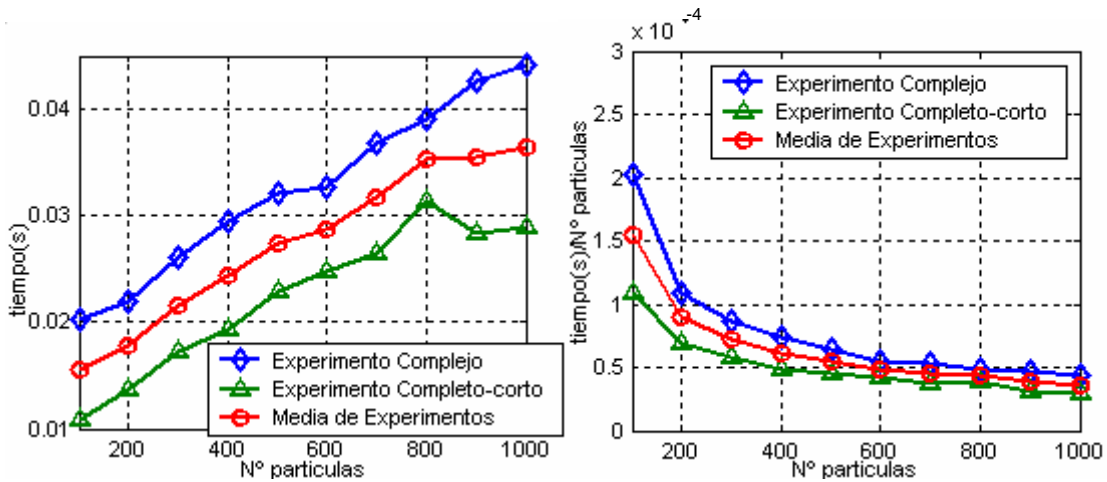


Figura 6.6. Gráficas comparativas del valor medio del t_{exe} del XPFCP (imagen izquierda) y del t_{exe} por partícula (imagen derecha) en función de n , en los mismos experimentos de la Figura 6.4.

A partir de los análisis anteriores se puede concluir que la propuesta de estimación multimodal de la tesis, tiene una fiabilidad independiente del número de partículas empleadas (a partir de un valor mínimo de $n=400$ aproximadamente en la aplicación de interés, según la gráfica derecha de la Figura 6.6) y de la complejidad del experimento. La razón de la robustez del algoritmo independientemente del valor de n y de la complejidad de la situación, se encuentra en el uso del clasificador de medidas $Y_t \Rightarrow G_{1:k,t|in}$ en sus etapas de re-inicialización y corrección, como se explica en puntos anteriores: el uso en el XPFCP de la información determinística que aporta este segmentador, compensa la necesidad presente en otros algoritmos basados en el PF de aumentar el número de partículas de la creencia $p(\bar{x}_t^{n-n_{m,t}} | \bar{y}_{1:t})$, o de calcular información de asociación computacionalmente más costosa que la determinística generada por el clasificador propuesto, para obtener un resultado fiable de estimación.

Gracias a este comportamiento del XPFCP, éste se puede ejecutar con un número de partículas suficientemente pequeño como para alcanzar la especificación de tiempo real² perseguida, sin detrimento de la fiabilidad de la estimación multimodal. Se resuelve así uno de los paradigmas planteados por la comunidad científica en la aplicación del PF a problemas de estimación, que ha dado lugar a propuestas como el APF ([Pitt99]), el EKPF o su versión para sistemas no lineales (el UPF –[Merwe01]–), el “*ICondensation*” ([Isard98b]), o la inserción de pasos MCMC ([Liu98]) en el lazo básico del filtro bayesiano, todas ellas analizadas en el capítulo 2, que debido a su complejidad son raramente usadas en tareas con requerimientos de tiempo real como la de interés.

Por otro lado, conviene destacar en este punto que, como se confirma en apartados posteriores, el parámetro n es el único que afecta de manera relevante al t_{exe} del XPFCP, lo que confirma la importancia de su correcta elección en aplicaciones en tiempo real.

La Figura 6.7 muestra el valor instantáneo de \hat{n}_{eff} generado por el XPFCP al ser ejecutado en el experimento complejo con los mismos valores de n de la Figura 6.5. A la vista de la Figura 6.7 se concluye que este valor no se modifica en gran medida ante cambios grandes del parámetro n : cambia una media de un 1.5% al modificar n en alrededor de un 50%. Esta conclusión confirma la idea expuesta en capítulos anteriores, de que el aumento del número de partículas usadas en el filtro bayesiano no es una solución al problema de empobrecimiento del algoritmo. En apartados posteriores se muestra cómo las mejoras al PF básico que dan lugar al XPFCP, y el ajuste de los parámetros de los que éstas dependen, sí aumentan de forma apreciable el valor de este factor de calidad \hat{n}_{eff} , y por ende eliminan el problema mencionado.

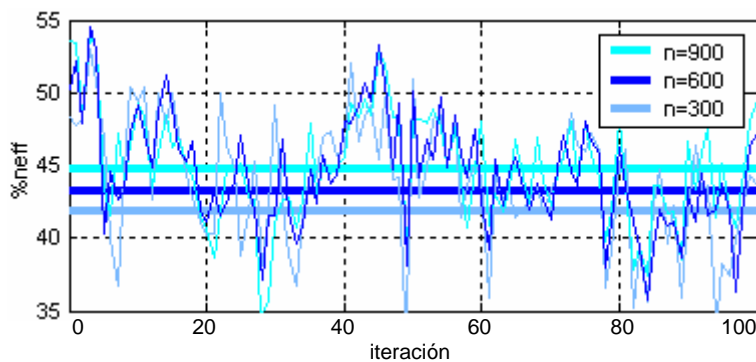


Figura 6.7. Gráficas comparativas del valor de \hat{n}_{eff} generado por el XPFCP en función de n , obtenidas en los mismos experimentos de la Figura 6.5. Las líneas de trazo grueso muestran el valor medio de las de trazo fino.

La Figura 6.8 muestra el valor medio de \hat{n}_{eff} obtenido de la ejecución del filtro en función de n , en las mismas condiciones de test de la Figura 6.4. La Figura 6.8 confirma la conclusión expuesta en el párrafo anterior, y muestra además que la ligera mejora que produce el aumento del número de partículas n del XPFCP en el valor del factor de calidad \hat{n}_{eff} sólo tiene relevancia en situaciones sencillas de estimación (gráfica verde), mientras que éste no mejora

² En los resultados temporales mostrados en la tesis se considera tiempo real a un procesado de 15FPS, como se comenta en el capítulo 2, pues los tests usados se corresponden con un t_s en la captura de imágenes de 15FPS.

con el aumento de n en situaciones complejas (gráfica azul). Se reafirma, por tanto, la idea de que el incremento del número de partículas usadas no resuelve el problema de empobrecimiento típico del PF y de sus versiones en su aplicación a la tarea de MTT objetivo de esta tesis.

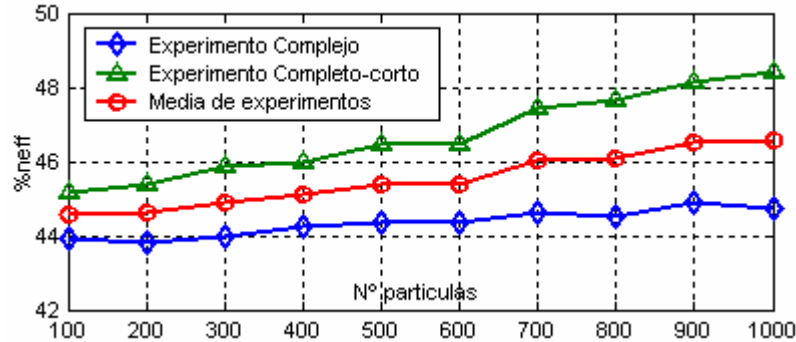


Figura 6.8. Gráficas comparativas del valor medio de \hat{n}_{eff} generado por el XPFCP en función de n , en los en los mismos experimentos de la Figura 6.4.

La unificación de las especificaciones de fiabilidad y tiempo real que se consiguen en el XPFCP, como se explica en párrafos anteriores, evita la necesidad de emplear técnicas de “*resampling selectivo*” como las expuestas en [Kreucher05] o [Gustafsson02] (ver en el capítulo 4), en función del valor de \hat{n}_{eff} que arroja el filtro en cada instante, que complican la funcionalidad del estimador y no aseguran su ejecución en tiempo real en todas las situaciones (el t_{exe} del PF con “*resampling selectivo*” es variable). Los resultados mostrados en este punto llevan a fijar como valor de ajuste óptimo del número de partículas $n_{opt} = 600$.

b) Efecto de la modificación de $n_{m,t-1} = \gamma_t \cdot n$

En este apartado se analiza el efecto que tiene en la funcionalidad del XPFCP la modificación del número de partículas relativas $\gamma_t = \frac{n_{m,t-1}}{n}$ insertada en su paso de re-inicialización. Como se explica en el apartado 6.1.1, esta relación se hace depender del número de clases de medidas nuevas generadas por el segmentador a partir del set $Y_{t-1} \Rightarrow G_{k,t-1|in}$ en cada instante (ver ecuación <6.15>). Para analizar el efecto del parámetro γ_t de forma independiente a esta circunstancia, en las pruebas incluidas a continuación se ha fijado constante el valor de $n_{m,t-1} = n_m$. En el apartado siguiente se analiza el efecto en la fiabilidad y robustez del XPFCP del factor que hace variable el parámetro $n_{m,t-1}$.

La Figura 6.9 muestra la tasa de error generada por el XPFCP en distintos experimentos (gráfica azul para el experimento complejo, verde para el completo corto, y roja con la media de los resultados obtenidos en ambas pruebas) y función de la relación γ_t .

La Figura 6.9 muestra la importancia del parámetro de relación normalizado γ_t . Su variación de 0.1 a 0.9 produce una desviación típica en la tasa de error de hasta el 8.5%, y una diferencia media en la tasa de error máxima respecto de la mínima de más de 17 puntos, y es

más importante en situaciones complejas de estimación. Un valor excesivamente bajo de γ_t (bajo número de partículas $n_{m,t-1}$) elimina el efecto del paso de re-inicialización, aumentando mucho la tasa de error del XPFCP, que no puede funcionar de forma multimodal sin él, como se explica en el capítulo 4. Por otro lado, un valor excesivo de γ_t (alto número de partículas $n_{m,t-1}$) hace que el filtro pierda su rigor matemático (como se corrobora más adelante en este punto), y por tanto su fiabilidad de estimación. La fiabilidad del filtro bayesiano permanece constante para valores intermedios del parámetro γ_t (entre 0.3 y 0.7 según la figura), no siendo necesario un valor elevado de éste para alcanzar el comportamiento óptimo del XPFCP.

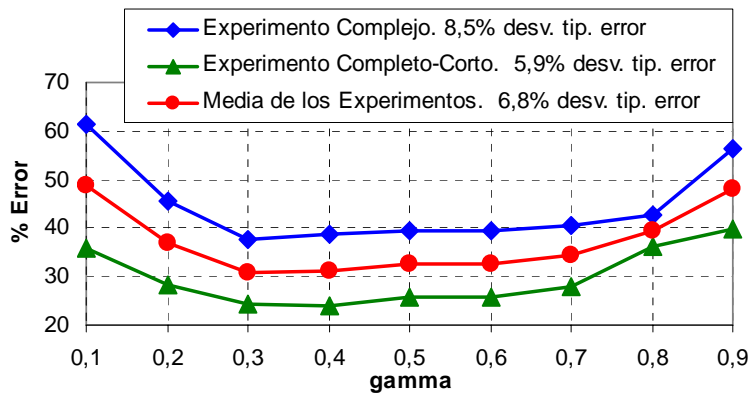


Figura 6.9. Tasa de error generada por el XPFCP en función del valor de $\gamma_t = \frac{n_{m,t-1}}{n}$ ($\text{gamma} = n_m/n$), en el experimento complejo (en azul) y en el completo corto (en verde), y valor medio de estos resultados (en rojo).

Por otro lado, como se expone en el punto anterior, los resultados empíricos demuestran que la modificación del parámetro γ_t no tiene efectos apreciables en el t_{exe} del algoritmo.

La Figura 6.10 muestra el valor instantáneo de \hat{n}_{eff} generado por el estimador en el experimento complejo, con distintos valores del ajuste fijo del parámetro γ_t ($\gamma_t = 0.2$ en la gráfica azul oscura, $\gamma_t = 0.1$ en la de color azul intenso y $\gamma_t = 0.9$ en la azul clara). En la figura se observa la relevancia que tiene el ajuste del parámetro γ_t en el valor del factor \hat{n}_{eff} generado por el XPFCP.

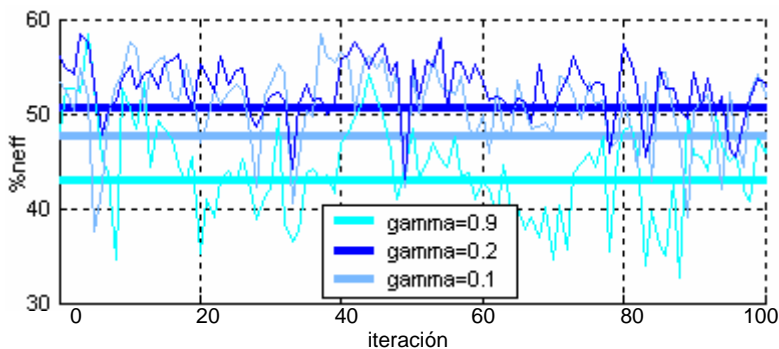


Figura 6.10. Gráficas comparativas del valor de \hat{n}_{eff} generado por el XPFCP en función del valor de γ_t (gamma), obtenidas en el experimento complejo. Las líneas de trazo grueso muestran el valor medio de las de trazo fino.

La Figura 6.11 confirma la conclusión expuesta en el párrafo anterior. En ella se muestra el valor medio de \hat{n}_{eff} en función del valor de γ_t y en las mismas condiciones de test de la Figura 6.9. Del análisis de la Figura 6.11 se puede concluir que, desde el punto de vista del factor de calidad \hat{n}_{eff} , el valor de ajuste más adecuado para el parámetro γ_t en el XPFCP es de 0.2, y que, como se observa también en la Figura 6.10, este factor de calidad disminuye notablemente según el valor de γ_t se aleja del óptimo (por cada modificación del parámetro en 0.1 puntos se pierde una media de 1.3 puntos en el valor de \hat{n}_{eff} respecto al mejor resultado) independientemente de la complejidad del experimento (del número de modos incluidos en la creencia discreta $p(\bar{x}_t^{n-n_{m,t}} | \bar{y}_{1:t})$ generada por el XPFCP en la aplicación de MTT de interés).

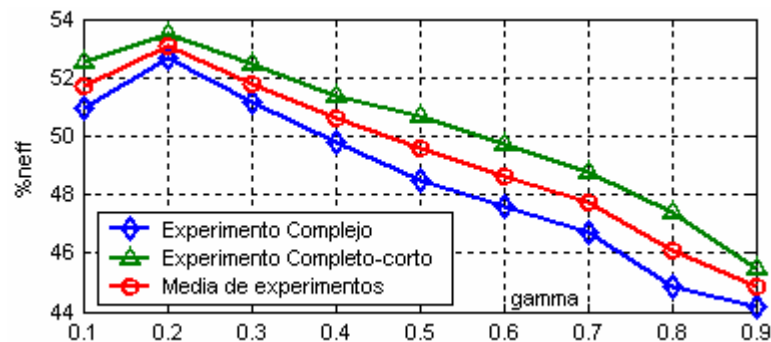


Figura 6.11. Gráficas comparativas del valor medio de \hat{n}_{eff} generado por el XPFCP en función del valor de γ_t (gamma), en los en los mismos experimentos de la Figura 6.9.

Merece la pena destacar que la disminución del valor medio de \hat{n}_{eff} que se hace más radical para valores altos de γ_t , según muestra la Figura 6.11, indica la degeneración del set de partículas del XPFCP, debida a la pérdida de rigor matemático del filtro. Como se comenta en puntos anteriores, al aumentar $\gamma_t = \frac{n_{m,t-1}}{n}$, y por ende el número de partículas renovadas en el paso de re-inicialización $n_{m,t-1}$, se disminuye a la vez el número de éstas que permanecen en el proceso iterativo bayesiano $n - n_{m,t-1}$, perdiéndose así la funcionalidad del filtro de partículas.

Se observa así que, al contrario de lo que ocurre con el número total de partículas n , el correcto ajuste del número de ellas que se introduce en el paso de re-inicialización del XPFCP ($n_{m,t-1}$, y por ende $\gamma_t = \frac{n_{m,t-1}}{n}$) tiene un efecto muy importante en el robustecimiento del filtro, y en la consiguiente eliminación del problema de divergencia y empobrecimiento del set (produce un aumento del factor de calidad \hat{n}_{eff} indicador de este hecho). Además se corrobora la inconveniencia de hacer excesivamente alto el número de partículas a incluir en este paso de re-inicialización (el valor de ajuste de $n_{m,t-1}$ debe estar en torno al 10% o 30% del número total de partículas n , es decir $\gamma_{opt,t} = [0.1, 0.3]$) pues esto no aporta fiabilidad a la estimación (como muestra la Figura 6.9) y sin embargo repercute en la pérdida del rigor matemático del filtro y en su consiguiente degeneración (como muestra la Figura 6.11).

c) Efecto de la modificación de $n_{m|j,t-1} / j = 1:k_{in,t-1}$ en función de la novedad de las clases de medidas

De los análisis expuestos en el apartado anterior se concluye que el valor óptimo de ajuste del número de partículas a insertar en el paso de re-inicialización $n_{m,t-1}$ ha de estar en un rango del 10% al 30% del número total n de ellas en el XPFCP. Como se explica en el apartado 6.1.2, el valor específico de este parámetro viene dado en cada instante t por el número de clases nuevas que se obtiene de la matriz de medidas Y_{t-1} , obtenida del sistema de sensado en $t - 1$.

En este apartado se analiza la ventaja que aporta esta consideración a la funcionalidad del filtro bayesiano propuesto, así como el efecto de la modificación del parámetro n_{init} , que especifica el número de partículas de más que se insertan por cada clase de medidas $G_{j,t-1|in} / j = 1:k_{in,t-1}$, en el instante t en caso de que ésta aparezca por primera vez en el instante $t - 1$ a la salida del clasificador (ver ecuación <6.15>).

La Figura 6.12 muestra la tasa de error del XPFCP en dos experimentos distintos (en azul la obtenida en el test complejo, en verde la obtenida en el completo corto y en rojo la media de ambas) en función de la modificación de $n_{m|j,t-1}$ debido al valor asignado al n_{init} .

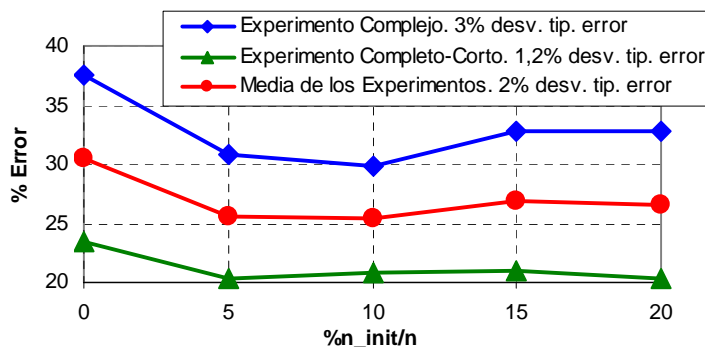


Figura 6.12. Tasa de error generada por el XPFCP en función del valor porcentual del parámetro n_{init} relativo al número total de partículas n usadas ($\% n_{init} / n$, $\%n_{init}/n$), en el experimento complejo (en azul) y en el completo corto (en verde), y valor medio de estos resultados (en rojo).

De la Figura 6.12 se desprende que el refuerzo de las hipótesis nuevas en el paso de re-inicialización mejora la fiabilidad del estimador en gran medida: la tasa de error generada por el XPFCP disminuye al incorporar el parámetro n_{init} en el cálculo de $n_{m|j,t-1}$ para todos los valores del parámetro probados, llegado a decrementarla en unos 5 puntos, de media en los dos experimentos mostrados, para el ajuste óptimo de n_{init} ; además, la disminución de la tasa de error del XPFCP que produce esta modificación es sobre todo evidente en situaciones complejas de estimación (en el experimento complejo la tasa de error disminuye en alrededor de 7 puntos), en las que es mayor la frecuencia de aparición y desaparición de hipótesis a considerar, o se produce un sensado más pobre de éstas, como demuestra el ejemplo descrito a continuación.

La Figura 6.13 muestra dos conjuntos de imágenes extraídos de la ejecución real del XPFCP en el experimento complejo de MTT sin refuerzo de las hipótesis nuevas (fila superior de imágenes) y con él (fila inferior de imágenes). De izquierda a derecha, las gráficas muestran: el peso normalizado de cada partícula tras el paso de corrección (set \tilde{w}_t); el número de “hijos” que obtiene cada una de ellas tras el de selección; y la representación de éste último set S_t , que constituye además el de salida del XPFCP, en la imagen de entrada al algoritmo. Los puntos rojos de las imágenes representan la proyección de la posición caracterizada por cada partícula en el espacio UV (gráficas izquierdas) y en el XZ (gráficas derechas y centrales), y los verdes la de las medidas obtenidas del sistema de sensado en el instante anterior Y_{t-1} .

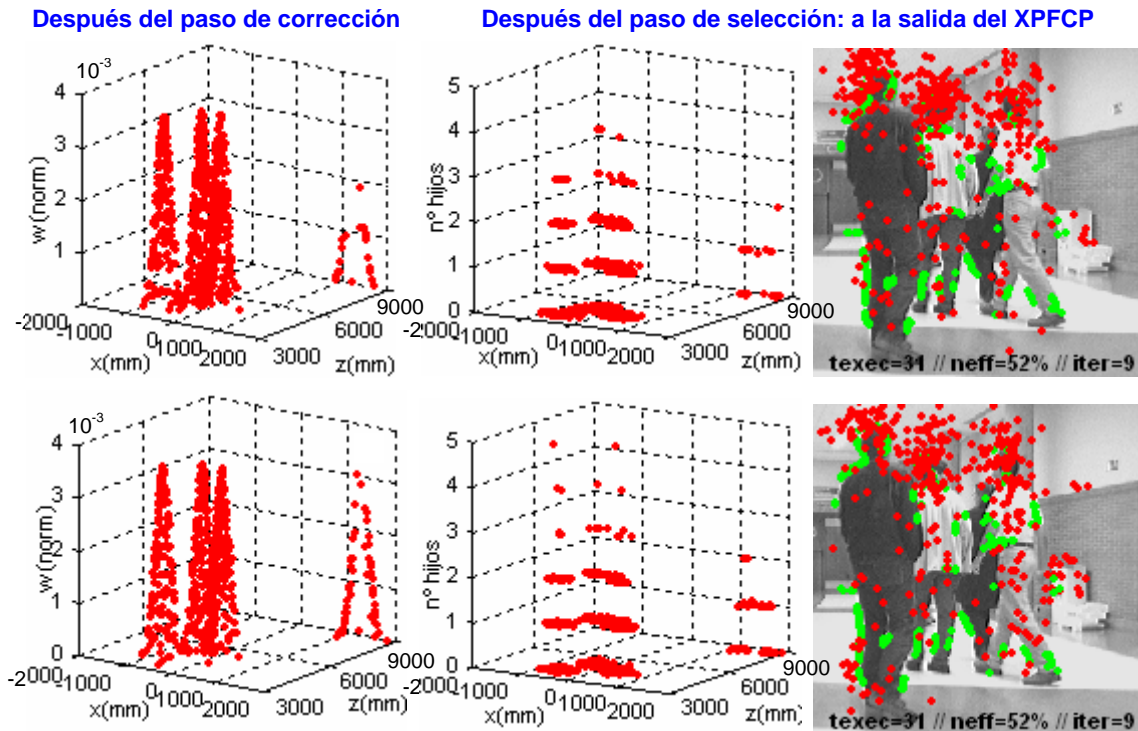


Figura 6.13. Imágenes extraídas de la evolución del set de partículas en la ejecución en tiempo real de la segunda parte del experimento complejo, del XPFCP sin (imágenes superiores) y con (imágenes inferiores) refuerzo de las hipótesis nuevas. La gráfica izquierda muestra el peso normalizado de cada partícula $\tilde{w}_t^{(l;n)}$ tras el paso de corrección; la central el número de “hijos” que obtiene cada una de ellas tras el de selección; y la derecha la representación de éste último set S_t (que constituye la salida del XPFCP) en la imagen de entrada. En todas las gráficas, los puntos rojos representan la proyección de cada partícula en el espacio UV (gráficas derechas) y en el XZ (gráficas izquierdas y centrales), y los verdes la de las medidas de Y_{t-1} en el instante anterior. Se incluye el t_{exe} y el valor de \hat{n}_{eff} (neff) generado por el XPFCP cada caso en el instante t .

En la Figura 6.13 se muestra una situación compleja de estimación multimodal debida a la aparición de un objeto nuevo en la escena (papelera al fondo y a la derecha en la imagen) al moverse la persona que lo mantenía oculto al sistema de observación. El conjunto de partículas $\hat{s}_{t-1}^{(i)} / i = 1 : n_{m|j1,t-1}$ insertado en la etapa de re-inicialización para caracterizar a este nuevo objeto en la PDF correspondiente es más numeroso si se incluye el refuerzo de la nueva hipótesis a

través del parámetro n_{init} (ver la imagen inferior izquierda, en la zona $XZ = ([1000,2000], [8000,9000])$) que si éste no se incluye (ver la imagen superior izquierda en la misma zona). Además en situaciones como ésta, las partículas insertadas en el paso de re-inicialización obtienen un peso menor al del resto en la etapa de corrección, independientemente del uso del proceso de refuerzo comentado, debido a la pobreza de las medidas relacionadas con el objeto nuevo (puntos verdes sobre la papelera en las imágenes).

Como consecuencia de lo expuesto en el párrafo anterior, el número de “hijos” generados para este set $\hat{s}_{t-1}^{(i)} / i = 1:n_{m|j_{1,t-1}}$ (ver la zona $XZ = ([1000,2000], [8000,9000])$ de las imágenes centrales de la Figura 6.13), por el proceso de resampling del XPFCP con refuerzo de las hipótesis nuevas es suficientemente alto como para representar de forma robusta la nueva hipótesis en la creencia discreta $p(\bar{x}_t^{n-n_{m,t}} | \bar{y}_{1:t})$ a la salida del estimador multimodal (ver imágenes derechas de la misma figura), incluso en situaciones complejas como la descrita. En este tipo de situaciones, la distribución equitativa de las $n_{m,t-1}$ partículas a insertar en el paso de re-inicialización, en función del conjunto de clases de medidas $Y_{t-1} \Rightarrow G_{1:k,t-1|in}$, no genera en la creencia de salida del XPFCP una representación suficientemente robusta de las hipótesis nuevas, de modo que ésta desaparece de la creencia en pocas iteraciones o “frames”³. Este hecho justifica los resultados mostrados en la Figura 6.12.

Del análisis de los resultados mostrados en la Figura 6.13 se obtienen los datos incluidos en la Tabla 6.1 que corroboran las conclusiones expuestas en los párrafos anteriores. Estos datos indican el reparto porcentual del número de partículas $n - n_{m,t}$ del set de salida del XPFCP S_t asociado a cada hipótesis de seguimiento implicada en el experimento de la figura⁴.

Tabla 6.1. Reparto porcentual del número de partículas del set de salida del XPFCP S_t asociado a cada una de las cuatro hipótesis de seguimiento del experimento de la Figura 6.13. Se indica el número porcentual de partículas de los objetos numerados según aparecen ubicados de izquierda a derecha en las imágenes derechas de la Figura 6.13.

Algoritmo/Objeto	1	2	3	4
$\frac{n^\circ \text{ Particulas Asociado}}{n - n_{m,t}} \times 100$ XPFCP con refuerzo	28.5	28.1	31.5	10.9
$\frac{n^\circ \text{ Particulas Asociado}}{n - n_{m,t}} \times 100$ XPFCP sin refuerzo	31.2	42.2	24.4	2.2

Los datos expuestos en la Tabla 6.1 muestran que el XPFCP con refuerzo permite obtener una representación más robusta de la hipótesis nueva en la creencia de salida $p(\bar{x}_t^{n-n_{m,t}} | \bar{y}_{1:t})$ que el filtro que no incluye este proceso. Además, el resultado mostrado en la tabla indica que el

³ El término inglés “frame” referencia a cada iteración del tracker realizada sobre una imagen, sobre un conjunto de datos de entrada diferente. Debido a su uso extendido en la literatura del área éste se emplea en la tesis sin traducir.

⁴ Este dato se ha obtenido analizando la posición que representa cada partícula en el espacio XYZ .

proceso de refuerzo equilibra la probabilidad de la representación de las cuatro hipótesis en esta PDF, que queda, por tanto, robustecida.

Por otro lado, la Figura 6.14 muestra el valor medio del factor de calidad \hat{n}_{eff} que arroja el XPFCP en las mismas condiciones de test de la Figura 6.12, en función del valor asignado al parámetro n_{init} . Los resultados mostrados en la Figura 6.14 indican que la modificación de este parámetro no cambia en exceso la robustez global de la representación discreta de la creencia a la salida del filtro $p(\bar{x}_t^{n-n_{m,t}} | \bar{y}_{1:t})$, si bien la incorporación de n_{init} en el cálculo de $n_{m|j,t-1}$ disminuye ligeramente (en una media de un 0.3%) el factor de calidad \hat{n}_{eff} ⁵. La mejora de la robustez obtenida con la inclusión del proceso de refuerzo es pequeña debido a que, como demuestra la Figura 6.13 este proceso sólo tiene un efecto apreciable en la robustez de la creencia discreta cuando se incorpora en ella alguna hipótesis nueva.

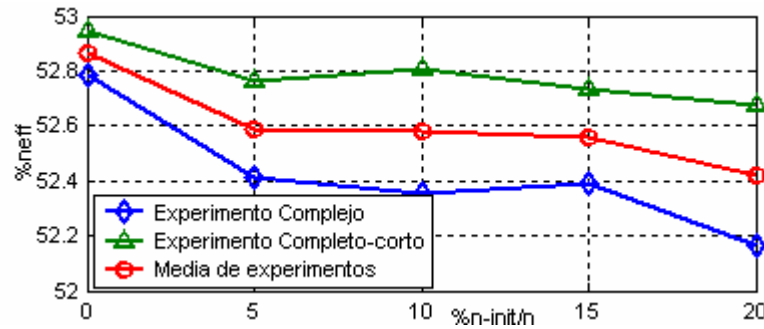


Figura 6.14. Gráficas comparativas del valor medio de \hat{n}_{eff} generado por el XPFCP en función del valor porcentual de n_{init} relativo a n ($\%n_{init}/n$, $\%n-init/n$), en los experimentos de la Figura 6.12.

El refuerzo con n_{init} partículas de cada hipótesis nueva detectada a través de las clases de medidas $G_{j,t-1|in} / j=1:k_{in,t-1}$ aumenta el valor de $n_{m,t-1} = \sum_{j=1}^{k_{in,t-1}} n_{m|j,t-1} = \gamma_t \cdot n$ según la expresión <6.15>. El ligero empobrecimiento en el set S_t de salida del XPFCP que produce el aumento del parámetro n_{init} (ver la Figura 6.14) equivale, por tanto, al que se produce con el aumento del parámetro γ_t y que se muestra en la Figura 6.11. El efecto de refuerzo de las hipótesis nuevas representadas a la salida de la re-inicialización por el set $\hat{S}_{t-1} = \{\bar{x}_{t-1}^{(i)}, 1/n\}_{i=1}^n$ se limita, por tanto, para evitar el conocido problema de empobrecimiento. Esto se consigue estableciendo un valor máximo $n_{m|max}$ para el número total de partículas a insertar $n_{m,t-1}$ en el paso de re-inicialización, como se muestra en la Figura 6.15.

A partir del rango de trabajo $\gamma_t = [0.1, 0.4]$, establecido por los resultados de la Figura 6.11, se determina el valor de ajuste de los parámetros de la Figura 6.15: $n_{m|max} = 0.4 \cdot n$ y

⁵ En este punto no se incluye una gráfica con el valor instantáneo de \hat{n}_{eff} , equivalente a las mostradas en apartados anteriores, debido a la similitud que presenta este factor de calidad para distintos valores del parámetro n_{init} .

$n_{m|\min} = 0.1 \cdot n$. Con objeto de que la relación entre $n_{m,t-1}$ y el número de clases de medidas nuevas detectadas en ese instante $t-1$ por el clasificador sea lineal en la mayor parte de los casos (zona sombreada de gráfica de la Figura 6.15, limitada por $k_{init} = (n_{m|\max} - n_{m|\min})/n_{init}$), se decide asignar al parámetro $n_{init,opt}$ un valor del 5% del número total de partículas n , consiguiéndose así mantener la relación lineal hasta en situaciones complejas en las que el segmentador detecte hasta 4 clases de medidas nuevas ($k_{init} = (0.3 - 0.1)/0.05 = 4$).

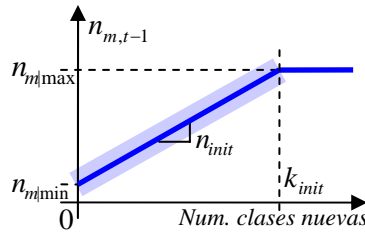


Figura 6.15. Representación gráfica de relación lineal limitada entre el número total de partículas a insertar en la re-inicialización del XPFCP $n_{m,t-1}$ y el de clases de medidas nuevas detectadas por el segmentador en Y_{t-1} .

Los ajustes comentados mejoran la fiabilidad del XPFCP y la robustez de la representación de las hipótesis que se incorporan en cada instante a la creencia de salida del XPFCP.

d) Efecto de la ponderación del set de pesos \bar{w}_t por la verosimilitud de las clases de medidas

En el apartado 6.1.1, se justifica de forma teórica que la incorporación de la verosimilitud $p_{j,t|in}$ de las clases de medidas $G_{j,t|in} / j = 1:k_{in,t}$ en el proceso de pesado de las partículas, a través de de la expresión <6.11>, mejora la robustez del paso de corrección y disminuye la posibilidad de degeneración del set de partículas S_t que caracteriza a la creencia discreta $p(\bar{x}_t^{n-n_{m,t}} | \bar{y}_{1:t})$ de salida del XPFCP. Este proceso robustece, en general, la representación de todas las hipótesis incluidas en esta PDF discreta, completando así el descrito en el apartado anterior que lo hace sólo con las nuevas. En este punto se demuestran empíricamente estas afirmaciones mediante pruebas semejantes a las ya expuestas.

La Figura 6.16 muestra la tasa de error del XPFCP en los experimentos complejo y completo corto, así como la media de los dos resultados, en función de la incorporación (en azul oscuro) o no (en azul claro) del valor de verosimilitud de las clases $p_{1:k,t|in}$ en el cálculo de los pesos. Como muestra la figura, el proceso de ponderación mejora la fiabilidad del filtro bayesiano, disminuyendo de media en 3.5 puntos la tasa de error que éste genera, siendo esta mejora más evidente en situaciones complejas de estimación (esta tasa se disminuye en 5 puntos en el experimento complejo al añadir al XPFCP el proceso de ponderación de pesos).

Por otro lado, la Figura 6.17 muestra el valor instantáneo del factor de calidad \hat{n}_{eff} generado en el experimento complejo por el XPFCP sin ponderación (en azul claro) y con ella (en azul

oscuro). La figura incluye una tabla con el valor medio numérico de este factor de calidad para los dos casos, en el mismo experimento ((a), mostrado también en trazo grueso del mismo color en la gráfica de la figura) y en el completo corto (b).

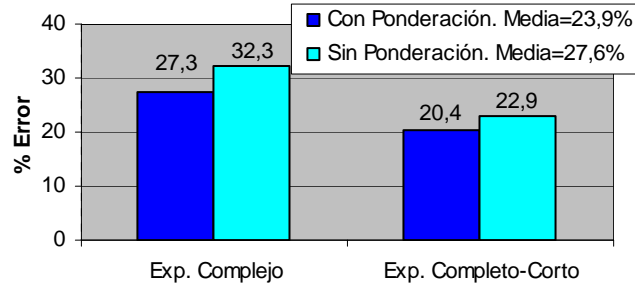


Figura 6.16. Tasa de error generada por el XPFCP en los experimentos complejo y completo corto, en función de la aplicación de ponderación a los pesos \vec{w}_t (con ponderación en azul oscuro y sin ponderación en azul claro), por la verosimilitud $p_{1:k,1:t|in}$ de la clase de medidas $G_{j,t|in} / j = 1:k_{in,t}$ a la que se asocia cada una.

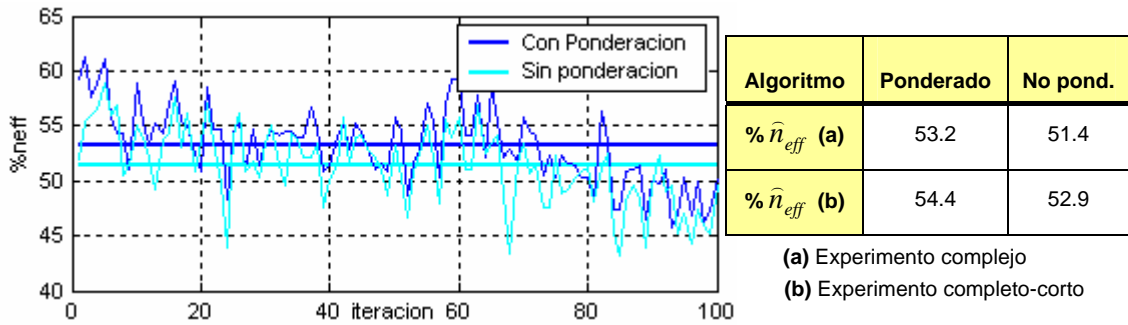


Figura 6.17. Gráficas comparativas del valor de \hat{n}_{eff} generado por el XPFCP en el experimento complejo en función de la ponderación o no del array de pesos \vec{w}_t por la verosimilitud $p_{1:k,1:t|in}$ de la clase de medidas $G_{j,t|in} / j = 1:k_{in,t}$ correspondiente. Las líneas de trazo grueso muestran el valor medio de las de trazo fino. Se incluye además una tabla con el valor medio de \hat{n}_{eff} en ese test (a) y en el completo corto (b).

La Figura 6.17 confirma empíricamente la mejora de robustez que el proceso de ponderación mencionado aporta al set de partículas S_t de salida del estimador multimodal propuesto, independientemente de la complejidad del experimento (en las dos pruebas se obtiene una mejora con este proceso de alrededor de un 1.5% en el valor del factor de calidad \hat{n}_{eff}).

Finalmente, la Figura 6.18 ejemplifica el efecto de la ponderación de pesos a través de unas imágenes extraídas del mismo experimento real presentado en la Figura 6.13, justificando así los resultados obtenidos en las figuras anteriores. El significado de las gráficas incluidas en esta Figura 6.18 es el mismo que el de las mostradas en la Figura 6.13.

El uso de la probabilidad validada $\frac{p_{j,1:t|in}}{p_{valid}} = p_{\min,i,1:t}$ de la clase de medidas $G_{j,t|in} / j = 1:k_{in,t}$ asociada a cada partícula $\vec{x}_t^{(i)} / i = 1:n$ en la ponderación de sus pesos

$\tilde{w}_t^{(i)} / i = 1:n$ hace que estos últimos tengan un valor proporcional a la fiabilidad de la clase asociada, tal y como se observa en la imagen izquierda de la Figura 6.18, en lugar del valor equilibrado que este set $\tilde{w}_t^{(i)} / i = 1:n$ presenta sin la ponderación (ver imágenes izquierdas de la Figura 6.13). El resultado de aplicar al conjunto de pesos así obtenido el proceso de resampling, tiene una consecuencia muy importante en el robustecimiento del set S_t a la salida de este último paso del XPFCP, y por ende, de la creencia multimodal que éste genera como solución al problema de estimación de interés $p(\bar{x}_t^{(1:n-n_{m,t})} | \bar{y}_{1:t})$: el número de muestras asociadas a cada hipótesis de estimación en esta creencia discreta final se reparte entre ellas de forma más equilibrada (sin diferencias muy relevantes) que si no se aplica la ponderación de partículas descrita (ver imagen central e izquierda en la Figura 6.18 en comparación con las mismas imágenes de la Figura 6.13.).

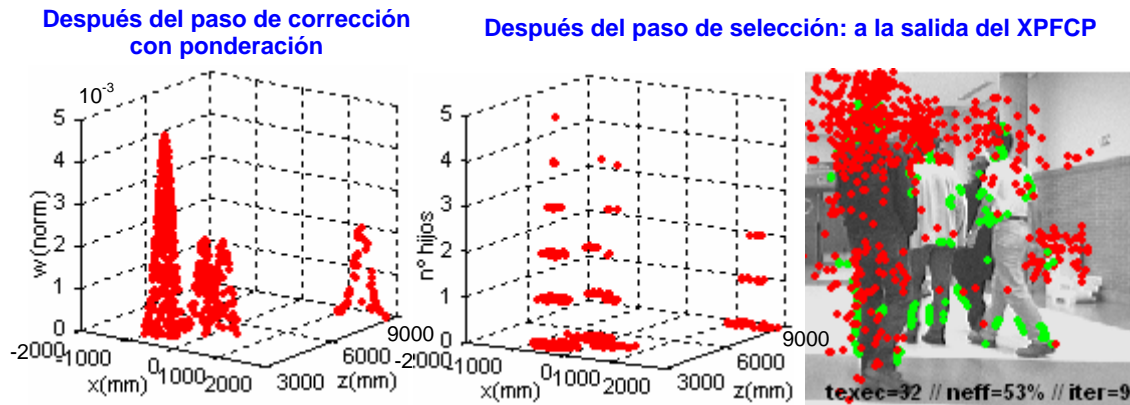


Figura 6.18. Imágenes extraídas de la evolución del set de partículas en la ejecución en tiempo real del XPFCP con proceso de ponderación de pesos, en la segunda parte del experimento complejo. El significado de las gráficas incluidas en la figura es el mismo que el de las incluidas en la Figura 6.13.

Los datos mostrados en la Tabla 6.2 representan la robustez en la creencia discreta final $p(\bar{x}_t^{(1:n-n_{m,t})} | \bar{y}_{1:t})$ de las cuatro hipótesis de seguimiento implicadas en el experimento de la Figura 6.18, de forma equivalente a los expuestos en la Tabla 6.1 sobre el test de la Figura 6.13. Los datos de la Tabla 6.2 corroboran lo expuesto en el párrafo anterior: todas las hipótesis tienen aproximadamente la misma robustez (excepto el objeto 1, mejor representado en el set de medidas Y_t) en lo que se refiere al número de partículas asociadas a cada una de ellas en el set de salida del XPFCP S_t , al añadirle el proceso de ponderación aquí descrito; además, la fiabilidad de cada una de las hipótesis es independiente de su novedad en la situación de tracking, y depende de la verosimilitud $p_{j,1:t|m}$ de la clase de medidas $G_{j,t|m} / j = 1:k_{in,t}$ con la que se relaciona, siendo esta última alta gracias al proceso de validación de estas clases.

Se completa así la tarea de robustecimiento que realiza el proceso de refuerzo del paso de re-inicialización, descrito en el punto anterior, sólo con las hipótesis nuevas.

Tabla 6.2. Reparto porcentual del número de partículas del set de salida del XPFCP S_t asociado a cada una de las cuatro hipótesis de seguimiento del experimento de la Figura 6.18. Se indica el número porcentual de partículas de los objetos numerados según aparecen ubicados de izquierda a derecha en las imágenes derechas de la Figura 6.13.

Algoritmo/Objeto	1	2	3	4
$\frac{n^\circ \text{ Particulas Asociado}}{n - n_{m,t}} \times 100$ XPFCP con ponderación	60.1	13.6	12.9	13.4

e) Efecto de los parámetros de ruido del modelo de actuación y de observación

En este punto se analiza la sensibilidad del XPFCP al correcto ajuste de los parámetros del modelo empleado en el proceso de estimación. Las ecuaciones de la expresión <6.18>, repetición de las anteriores <2.6> y <2.8> del capítulo 2 son, respectivamente, las ecuaciones de actuación y observación del modelo CV propuesto para la tarea de estimación de interés.

$$\begin{aligned} \bar{x}_t &= A \cdot \bar{x}_{t-1} + \bar{v}_{t-1} \\ \bar{y}_t &= C \cdot \bar{x}_t + \bar{o}_t \end{aligned} \quad <6.18>$$

donde $A = \begin{bmatrix} 1 & 0 & 0 & t_s & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & t_s \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ es la matriz de transición del sistema que sólo depende del

periodo de muestreo del estimador discreto t_s , $C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ es la matriz de

observación, y $\bar{v}_t = [v_{x,t} \ v_{y,t} \ v_{z,t} \ v_{vx,t} \ v_{vz,t}]^T$ y $\bar{o}_t = [o_{x,t} \ o_{y,t} \ o_{z,t}]^T$ son los vectores de ruido de estado y de medida respectivamente.

Gracias a la generalidad del modelo CV empleado sólo aparecen, por tanto, dos parámetros cuya modificación puede afectar a la fiabilidad de la estimación: el modelo del ruido de estado \bar{v}_t y el del ruido de medida \bar{o}_t . El vector \bar{v}_t se emplea en el proceso de predicción del valor de las partículas (en la etapa de predicción del XPFCP), mientras que el segundo \bar{o}_t se usa en el cálculo del array de pesos \bar{w}_t de las partículas (en la etapa de corrección) a través de las ecuaciones <6.10> y <6.11>.

Dado que en el modelo CV el vector de estado $\bar{x}_t = [x_t \ y_t \ z_t \ vx_t \ vz_t]^T$ no directamente accesible a través de las medidas $\bar{y}_t = [x_t \ y_t \ z_t]^T$, tampoco lo es el de ruido asociado a éste \bar{v}_t , por lo que no es posible obtener un modelo para este ruido a través de análisis empíricos. Las pruebas de sensibilidad del XPFCP frente a las características de \bar{v}_t se hacen suponiendo que este ruido es gaussiano, blanco y de media nula (suposición generalmente adoptada por la comunidad científica, [Bar-Shalom93]) y modificando la varianza $\sigma_{v,i} / i = \{x, y, z, vx, vz\}$ de las componentes de \bar{v}_t según su significado físico.

En este sentido es necesario destacar que, debido a la funcionalidad del vector de ruido \vec{v}_t en el modelo CV (ecuación superior en <6.18>), las componentes $\sigma_{v,i}$ representan la desviación típica de la velocidad y aceleración del modelo de posición, tal y como se explica en el capítulo 2. El valor máximo de la desviación típica $\sigma_{v,i}$ debe corresponderse, por tanto, con la máxima aceleración esperada para los objetos implicados en la aplicación de tracking en interiores de interés. Este valor puede alcanzar $1m/s^2$ ⁶, si bien parece conveniente probar con valores menores, más habituales en estas aplicaciones, teniendo en cuenta además que esta variable de ruido afecta a la dispersión de las partículas $S_{t|t-1} = \left\{ \bar{x}_{t|t-1}^{(i)}, 1/n \right\}_{i=1}^n$, a la salida de la etapa de predicción del XPFCP, y con ello a la exactitud de la estimación, como se ve en la Figura 6.2.

En lo referente al modelo del ruido de medida \vec{o}_t , y aprovechando que el vector de medida e posición \vec{y}_t al que está asociado es directamente accesible a través del sistema de observación elegido, conviene realizar un estudio empírico sobre su magnitud previo al análisis de la sensibilidad del XPFCP frente a ésta. Este análisis empírico indica que el ruido de medida de posición \vec{o}_t , puede modelarse gaussiano y blanco, y con una matriz de covarianza O , consecuentemente diagonal, cuyas componentes no nulas tienen un valor que oscila en el intervalo $\sigma_{o,i} = [40,230]mm / i = \{x, y, z\}$.

La Figura 6.19 muestra la modificación de la tasa de error generada por el XPFCP en los experimentos complejo (en azul) y completo corto (en verde), así como la media de los dos resultados (en rojo), en función del valor asignado a la desviación típica $\sigma_{v,i}$ de las cinco componentes del vector de ruido de estado \vec{v}_t (de la ecuación <6.18>), supuestas gaussianas, incorreladas y de media nula. Los valores de $\sigma_{v,i}$ empleados en esta prueba van de $10mm$ a $400mm$ ⁷, en concordancia con lo comentado en párrafos anteriores.

De la Figura 6.19 se desprende que la sensibilidad del algoritmo frente a modificaciones de este parámetro no es excesivamente alta (una modificación del 100% del parámetro $\sigma_{v,i}$ produce un cambio de alrededor del 3% en la tasa de error), pero conviene tenerla en cuenta a la hora de realizar el ajuste del parámetro $\sigma_{v,i}$. El algoritmo presenta un máximo de fiabilidad $\sigma_{v,i} = 100mm$, por lo que el ajuste de este parámetro debe ser próximo a este valor.

Por otra parte, la Figura 6.20 muestra el valor instantáneo del factor de calidad \hat{n}_{eff} generado por el XPFCP en el experimento complejo para distintos valores del parámetro del modelo del ruido de estado $\sigma_{v,i}$ (azul intenso para $\sigma_{v,i} = 200mm$, azul oscuro para $\sigma_{v,i} = 100mm$ y azul

⁶ Suponiendo una velocidad máxima típica de este tipo de aplicaciones de $3m/s$, el cambio de velocidad máximo se ha obtenido asumiendo una parada del objeto en $1s$ a partir de ésta. La desviación típica de esta velocidad es, en ese caso, aproximadamente $3 \cdot \sigma_{i,v} \approx 3m/s / 1s$.

⁷ El estudio aquí incluido sobre el ajuste de $\sigma_{v,i}$ se hace en términos de posición, si bien la magnitud de esta variable no es la misma para todas sus componentes $i = \{x, y, z, vx, vz\}$.

claro para $\sigma_{v,i} = 10mm$). La figura incluye además una tabla con el valor medio numérico de \hat{n}_{eff} en función del valor asignado a $\sigma_{v,i}$ en el mismo experimento ((a), mostrado en trazo grueso con los mismos colores en función de $\sigma_{v,i}$ en la gráfica) y en el completo corto (b).

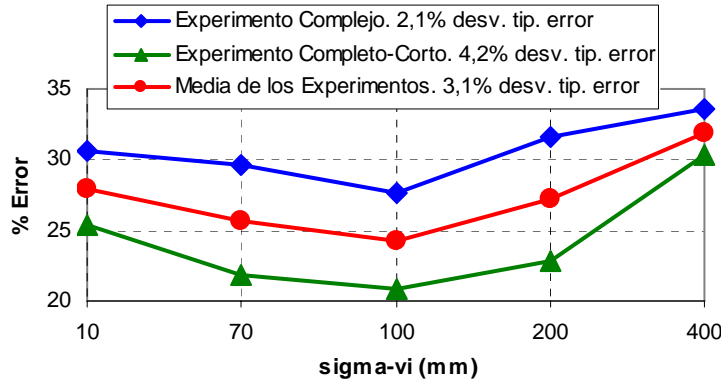


Figura 6.19. Tasa de error generada por el XPFCP en función de la desviación típica ($\sigma_{v,i}$, sigma-vi) de las componentes del vector de ruido de estado \vec{v}_t , supuestas gaussianas, incorreladas y de media nula, en el experimento complejo (en azul) y en el completo corto (en verde), y valor medio de ambos resultados (en rojo).

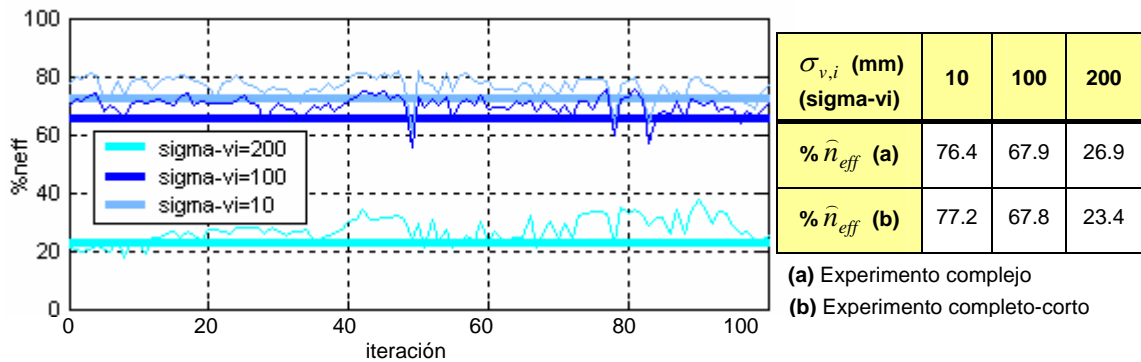


Figura 6.20. Gráficas comparativas del valor de \hat{n}_{eff} generado por el XPFCP en el experimento complejo, y en función de la desviación típica ($\sigma_{v,i}$, sigma-vi) de las componentes del vector de ruido de estado \vec{v}_t , supuestas gaussianas, incorreladas y de media nula. Las líneas de trazo grueso muestran el valor medio de las de trazo fino. Se incluye además una tabla con el valor medio de \hat{n}_{eff} en ese test (a) y en el completo corto (b).

La Figura 6.20 muestra la sensibilidad funcional del XPFCP del valor de este parámetro del ruido \vec{v}_t que afecta al vector de estado en el modelo CV elegido para el proceso de estimación. Valores altos del parámetro $\sigma_{v,i}$ hacen que el set de partículas S_t se empobrezca en gran medida (disminuyen en el factor de calidad \hat{n}_{eff} de la estimación), puesto que provoca la dispersión de las mismas ($\sigma_{v,i}^2$ representa la varianza de la velocidad del target en el modelo CV), y la consiguiente ineficiencia e inexactitud del proceso.

Por otro lado, la tabla incluida en la Figura 6.20 muestra que la conclusión expuesta en el párrafo anterior es independiente de la complejidad del experimento (es decir, del número de hipótesis de seguimiento o modos incluidos en la creencia discreta generada $p(\vec{x}_t^{n-n_{m,t}} | \vec{y}_{1:t})$ por

el XPFCP en la aplicación de MTT de interés), razón por la que no se ha incluido una gráfica con el valor medio de \hat{n}_{eff} en función de $\sigma_{v,i}$, para los dos experimentos, como se ha hecho en el análisis de otros parámetros en apartados anteriores.

A la vista de los comentarios expuestos se concluye que la funcionalidad del XPFCP es bastante sensible al modelo del ruido \vec{v}_t , supuesto éste blanco, gaussiano y de media nula. Además, se establece, un valor de ajuste de la desviación típica de las componentes de este ruido $\sigma_{v,i,opt} = 100mm$, que permite alcanzar máxima fiabilidad en los resultados generados por el XPFCP y un valor del factor de calidad \hat{n}_{eff} por encima del mínimo recomendado ([Gustafsson02]) para asegurar la eliminación del problema de empobrecimiento comentado ($\hat{n}_{eff|sigma-vi=100} = 69.85\% > 66\%$).

Al margen del análisis teórico sobre el modelo del ruido de medida \vec{o}_t , expuesto al principio de este punto, se incluye aquí también uno empírico que demuestra la validez del anterior. Este estudio permite además obtener un ajuste más fundamentado del valor de la desviación típica $\sigma_{o,i}$ de las tres componentes de este vector de ruido \vec{o}_t , supuesto gaussiano, blanco y de media nula, similar al realizado para el ruido \vec{v}_t que afecta al vector de estado.

La Figura 6.21 muestra la tasa de error generada por el XPFCP en función del valor del parámetro $\sigma_{o,i}$ en las mismas condiciones de la Figura 6.19. A partir de los análisis teóricos referidos, se usa como valor central del estudio una desviación típica de $\sigma_{o,i} = 100mm$.

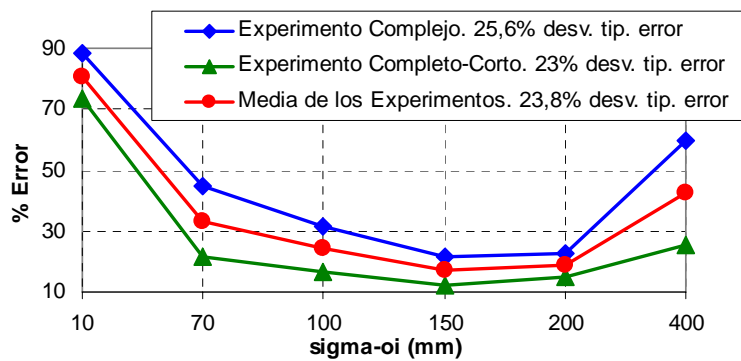


Figura 6.21. Tasa de error generada por el XPFCP en función de la desviación típica ($\sigma_{o,i}$, sigma-oi) de las componentes del vector de ruido de medida \vec{o}_t supuestas gaussianas, incorreladas y de media nula, en los mismos experimentos de la Figura 6.19.

La Figura 6.21 muestra que la sensibilidad del estimador bayesiano es mayor respecto al modelo del ruido de medida (al valor de $\sigma_{o,i}$, con una desviación típica media de casi el 30%, como muestra esta figura) que al de estado (valor de $\sigma_{v,i}$, con una desviación típica media 10 veces menor, del 3% en la Figura 6.19) para el mismo rango de estudio del parámetro. Este hecho se debe al mayor efecto que el paso de corrección de las partículas tiene respecto al de predicción en la fiabilidad del XPFCP, pues afecta a su ponderación y su posterior selección e inclusión en el proceso iterativo de filtrado.

La conclusión sobre la sensibilidad del XPFCP respecto al modelo del ruido de observación, expuesta en el párrafo anterior, confirma el hecho de que el uso de los centroides $\bar{g}_{1:k,t|in}$ de las clases de medidas $Y_t \Rightarrow G_{1:k,t|in}$, en lugar de las propias medidas, en el cálculo de este array de pesos \bar{w}_t (ver ecuaciones <6.10> y <6.11>) robustece el funcionamiento del estimador, pues filtra este ruido \bar{o}_t (disminuyendo $\sigma_{o,i}$) y minimiza su efecto en la fiabilidad del estimador.

En la Figura 6.21 se observa también que la sensibilidad del XPFCP frente a $\sigma_{o,i}$ es, al igual que frente a $\sigma_{v,i}$, independiente de la complejidad del experimento, si bien ésta se acentúa para valores altos de la desviación típica en experimentos complejos: en esos casos, al representar a más hipótesis de seguimiento, la degeneración del set de partículas S_t produce más errores.

En cualquier caso, a la vista de la Figura 6.21, es evidente que la fiabilidad del XPFCP en función del valor de $\sigma_{o,i}$ es máxima si éste se ajusta a $150mm$, lo que coincide con el aproximado para este parámetro en el análisis teórico incluido al principio de este punto. Se evidencia así la necesidad de ajustar el modelo del ruido de medida de acuerdo con el proceso de observación empleado en la tarea de MTT a resolver con el XPFCP. Este punto se pone de manifiesto en los resultados mostrados al final de este capítulo, en el que el filtro bayesiano propuesto se aplica a una situación de sensado diferente.

Por otro lado, la Figura 6.22 muestra el valor instantáneo del factor de calidad \hat{n}_{eff} que genera el filtro bayesiano en el experimento complejo, para los valores intermedios del parámetro $\sigma_{o,i}$ incluidos en el análisis anterior (azul intenso para $\sigma_{o,i} = 200mm$, rojo para $\sigma_{o,i} = 150mm$, azul oscuro para $\sigma_{o,i} = 100mm$ y azul claro para $\sigma_{o,i} = 10mm$). La figura incluye una tabla con el valor medio de \hat{n}_{eff} , en las mismas condiciones que la mostrada en la Figura 6.20. La tabla muestra que se obtienen resultados de \hat{n}_{eff} muy similares en los dos experimentos, razón por la que, de nuevo, no se incluyen en otra gráfica a parte.

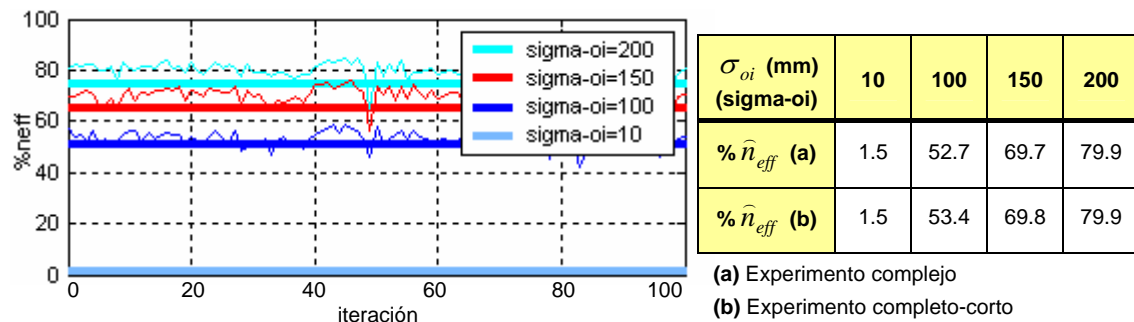


Figura 6.22. Gráficas comparativas del valor de \hat{n}_{eff} generado por el XPFCP en el experimento complejo, y en función de la desviación típica ($\sigma_{o,i}$, sigma-oi) de las componentes del vector de ruido de medida \bar{o}_t , supuestas gaussianas, incorreladas y de media nula. Las líneas de trazo grueso muestran el valor medio de las de trazo fino. Se incluye una tabla con el valor medio de \hat{n}_{eff} en ese test (a) y en el completo corto (b).

A la vista de la Figura 6.22 se concluye que el set de partículas S_t a la salida del XPFCP se robustece al aumentar este parámetro característico del modelo del ruido de medida. Este hecho se debe a que al aumentar el valor de $\sigma_{o,i}$ se incrementa la permisividad del proceso de pesado, por lo que se robustece el funcionamiento del filtro, y esto se refleja en un aumento \widehat{n}_{eff} .

Aunando las conclusiones sobre el efecto del valor del parámetro $\sigma_{o,i}$ en el XPFCP, obtenidas de la Figura 6.22 y la Figura 6.21, éste se ajusta, para las condiciones de sensado analizadas en este punto, a un valor de $\sigma_{o,i,opt} = 150mm$, maximizándose así la fiabilidad del estimador y manteniendo la riqueza del set que evita su degeneración ($\widehat{n}_{eff|sigma=oi=150} = 69.8\% > 66\%$).

f) Efecto del algoritmo de resampling

Tal y como se explica en el capítulo 4, y se justifica en [Merwe01], el uso de uno u otro algoritmo para desarrollar el paso de selección en el PF no aporta una mejora significativa en la eficiencia del estimador. En este apartado se pretende comprobar esta afirmación, testeando en el XPFCP tres de las propuestas de resampling más extendidas entre la comunidad científica por su baja carga computacional: el algoritmo multinomial, el residual y el sistemático (ver el apartado 4.2.1 del capítulo 4). Las distintas propuestas de selección presentadas en la literatura del área tienen como objetivo principal no modificar en gran medida ni la varianza del set $S'_t = \{\tilde{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ ni el coste computacional que presenta el filtro sin la inclusión de esta etapa, por lo que la elección de la propuesta más adecuada para el XPFCP en su aplicación a la tarea de MTT se obtiene del análisis de estos factores junto al de fiabilidad.

La Figura 6.23 muestra la tasa de error generada por el XPFCP en distintos experimentos, y en función del algoritmo de resampling empleado: multinomial (en azul intenso), residual (azul oscuro) o sistemático (azul claro). A la vista de la figura se concluye que, efectivamente, el uso de uno u otro algoritmo de re-muestreo no conlleva una diferencia sustancial en lo que a la fiabilidad de la estimación se refiere, independientemente, además de la complejidad del experimento (del número de hipótesis de seguimiento existentes en la aplicación de MTT de interés), si bien se obtiene una menor tasa de error con el residual.

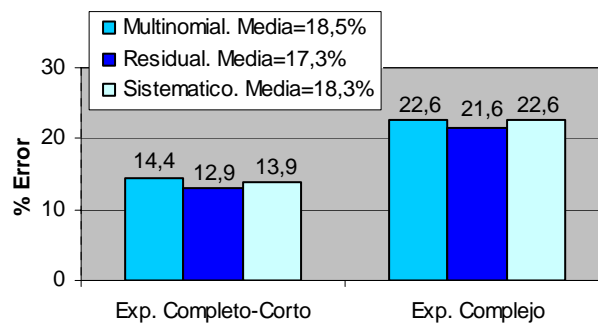


Figura 6.23. Tasa de error generada por el XPFCP en función del algoritmo de resampling empleado (multinomial en azul intenso, residual en azul oscuro y sistemático en azul claro), en los experimentos complejo y completo corto.

Por otro lado, el t_{exe} del XPFCP en las pruebas realizadas no es significativamente diferente en función del algoritmo de re-muestreo incluido, como cabía esperar, por lo que se concluye que este factor tampoco es determinante a la hora de decidir cuál de los tres procesos de resampling analizados conviene usar en la propuesta de filtrado probabilístico.

Para realizar estas pruebas, se modifican los algoritmos de resampling multinomial y sistemático con objeto de que generen $n - n_{m,t}$ partículas tal y como establece la funcionalidad del XPFCP, y de forma semejante a la expuesta en el apartado 6.1.1 para el algoritmo residual.

Finalmente, con objeto de analizar la diferente modificación que estos procesos de re-muestreo producen en la varianza del set de partículas $S_t^i = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$, y la aportación que producen en la mejora del recurrente problema de su empobrecimiento, se estudia el valor medio del factor de calidad \hat{n}_{eff} que se obtiene al emplear cada uno de ellos en los experimentos complejo y completo corto, obteniéndose los resultados que se muestran en la Tabla 6.3. La tabla muestra que tampoco se observa en este punto una diferencia importante (razón por la que se ha incluido una tabla en lugar de la gráfica de apartados anteriores), si bien el mejor resultado se consigue de nuevo con el algoritmo residual.

Tabla 6.3. Valor medio de \hat{n}_{eff} generado por el XPFCP en función del algoritmo de resampling usado en los experimentos complejo y completo-corto.

Test/Tipo resampling	Multinomial	Residual	Sistemático
% \hat{n}_{eff} · Complejo	69.4	69.9	69.7
% \hat{n}_{eff} · Completo Corto	69.7	69.8	69.8

A partir de las conclusiones expuestas se decide emplear el algoritmo de resampling residual debido a su sencilla implementación y a la ligera mejora en fiabilidad (mostrada en la Figura 6.23) y robustez (mayor valor del factor \hat{n}_{eff} mostrada en la Tabla 6.3, y por ende menor empobrecimiento del set de partículas) que aporta al XPFCP en su aplicación a la tarea de MTT.

g) Efecto del algoritmo de clasificación de medidas

En apartados anteriores de este capítulo, así como en el 5.4.1 del anterior, se justifica teóricamente la mayor idoneidad del “*k-medias secuencial*” con validación para desarrollar en los pasos de re-inicialización y corrección del XPFCP las tareas descritas. En este punto se incluye un análisis empírico de las ventajas que el uso de uno u otro clasificador de medidas $Y_t \Rightarrow G_{1:k,t|in}$ aporta a la fiabilidad, robustez y t_{exe} del filtro bayesiano propuesto en la tesis.

La Figura 6.24 muestra la tasa de error obtenida en los experimentos complejo y completo corto de MTT, en función de cuál de las dos propuestas de clasificación de medidas se combina con el filtro probabilístico para dar lugar al XPFCP. A la vista de esta figura, se concluye que la combinación del estimador multimodal descrita en puntos anteriores con el “*k-medias secuencial*” con validación, da lugar a una versión del XPFCP más fiable que la de éste con el

“*subtractive enlazado*” con identificación y validación, lo que coincide con la conclusión teórica expuesta en el capítulo 5.

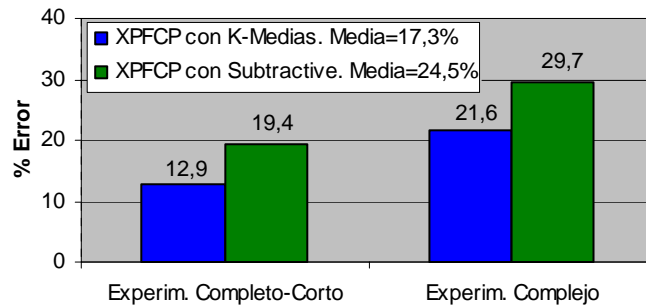


Figura 6.24. Tasa de error generada por el XPFCEP en función del algoritmo de clasificación de medidas usado (“*k-medias secuencial*” con validación en azul, o “*subtractive enlazado*” con identificación y validación en verde), en el experimento complejo y en el completo corto.

Por otro lado, se incluye también en este punto un análisis de la influencia que la elección de una u otra técnica de clasificación tiene en el t_{exe} del XPFCEP. Al contrario de lo que ocurre con el resto de parámetros analizados en este apartado, esta decisión sí influye en la carga computacional global del filtro bayesiano, debido a la importante diferencia que existe en el t_{exe} de las dos propuestas de clasificación presentadas en el capítulo 5 (ver allí punto 5.4.3).

La Figura 6.25 muestra el t_{exe} en el experimento complejo del XPFCEP en función del clasificador de medidas empleado (en azul oscuro con el “*k-medias secuencial*” con validación y en verde oscuro con el “*subtractive enlazado*” con identificación y validación). La figura muestra también, el t_{exe} únicamente del proceso de clasificación, en los mismos colores pero en tono más claro (azul claro para el clasificador booleano y verde claro para el borroso).

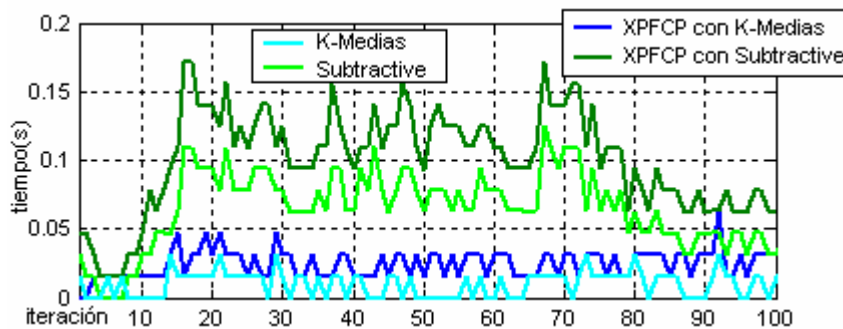


Figura 6.25. Gráficas comparativas del t_{exe} del XPFCEP en función del algoritmo de clasificación de medidas usado (con “*k-medias secuencial*” con validación en azul oscuro, y el del clasificador en azul claro, o con “*subtractive enlazado*” con identificación y validación en verde oscuro y el de el segmentador en claro), en el test complejo.

Coincidiendo con los análisis de los clasificadores incluidos en el capítulo 5, la Figura 6.25 muestra que el XPFCEP ve incrementado su t_{exe} si incluye el segmentador borroso, respecto a si se usa el booleano. De hecho, como muestra la figura, el t_{exe} del clasificador borroso (en verde claro) es bastante superior al global del XPFCEP basado en el booleano (azul oscuro).

La Tabla 6.4 muestra además, que de media el t_{exe} del filtro es casi 4 veces mayor al usar el “*subtractive enlazado*” en el XPFCEP que si se emplea el “*k-medias secuencial*”. En esta tabla

se incluye también el incremento de t_{exe} que produce la inclusión del algoritmo de clasificación en el filtro bayesiano en el mismo test, observándose que al emplear el segmentador borroso su coste computacional se incrementa mucho más (en un 82%) que con el booleano (en un 48%).

Tabla 6.4. Valor medio e incremento absoluto y relativo del t_{exe} del XPFCP al ejecutarlo en el experimento complejo, en función del algoritmo de clasificación de medidas usado.

Clasificador de medidas	t_{exe} total (s)	t_{exe} clasificador (s)	% t_{exe} clasificador del total
“K-Medias”	0.027	0.013	48%
“Subtractive”	0.100	0.082	82%

Es importante recordar que, como se expone en el apartado 5.4.3 del capítulo 5, el t_{exe} del clasificador depende del número de datos a organizar m_i en el caso del “*subtractive enlazado*”, y del número de clases k_i en el del “*k-medias secuencial*”, por lo que la diferencia en el t_{exe} del XPFCP será aún mayor, independientemente de la complejidad de la situación de MTT (que sólo afecta a k_i), si el set de datos de entrada a éste es denso (mayor m_i).

Finalmente, la Figura 6.26 muestra el valor de \hat{n}_{eff} generado por el XPFCP en función del tipo de clasificador de medidas usado, en las mismas condiciones de test de la Figura 6.25. La Figura 6.26 incluye una tabla, similar a la de la Figura 6.20, que recoge el valor medio de \hat{n}_{eff} en el mismo experimento ((a), mostrado también en trazo grueso del mismo color en la gráfica de la Figura 6.26) y en el completo corto (b).

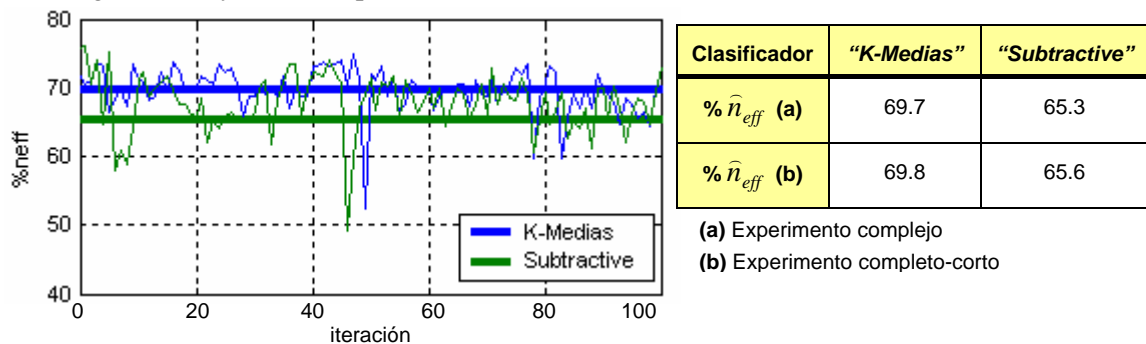


Figura 6.26. Gráficas comparativas del valor de \hat{n}_{eff} generado por el XPFCP en función del clasificador de medidas usado y en los experimentos de la Figura 6.25. Las líneas de trazo grueso muestran el valor medio de las de trazo fino. Se incluye una tabla con el valor medio de \hat{n}_{eff} en ese test (a) y en el completo corto (b).

A la vista de la Figura 6.26 se concluye que el uso del clasificador booleano en el XPFCP aporta también mayor robustez al conjunto de partículas, minimizando así el problema de empobrecimiento que éste puede sufrir, que la que aporta el borroso. El valor medio final que genera el XPFCP para este factor de calidad \hat{n}_{eff} , una vez realizado el ajuste de todos sus parámetros queda, por tanto, por encima del valor mínimo recomendado en la literatura del área para asegurar la eliminación del problema de empobrecimiento característico de los estimadores basados en el PF: $\hat{n}_{eff} = 69.8\% > 66\%$.

Los comentarios sobre los resultados expuestos a lo largo de este apartado llevan a la conclusión de que el “*k-medias secuencial*” con validación tiene un comportamiento más adecuado para llevar a cabo el proceso de clasificación de medidas $Y_t \Rightarrow G_{1:k,t|in}$, necesario en el funcionamiento del XPFCP como algoritmo de seguimiento multi-objeto.

A partir de esta conclusión, en el punto siguiente se analiza finalmente el ajuste de los parámetros del clasificador, utilizando como punto de partida del análisis el estudio de sensibilidad del “*k-medias secuencial*” con validación incluido en el capítulo 5.

h) Ajuste de los parámetros del algoritmo de clasificación de medidas

Para asegurar la fiabilidad de las funciones que realiza el “*k-medias secuencial*” como clasificador del set de medidas en el XPFCP, presentadas en el apartado 6.1.1, es necesario analizar el efecto que tiene en ella el correcto ajuste de sus parámetros.

En el apartado 5.2.6 del capítulo 5, se realiza un análisis de la sensibilidad de este clasificador respecto al valor de sus parámetros. Dicho estudio puede emplearse como base del análisis incluido a continuación pues se realiza usando como conjunto de datos a clasificar el set Y_t implicado en la segmentación de interés en este punto. No es posible, sin embargo, usar directamente ese estudio en el análisis del algoritmo como segmentador de partículas a la salida del XPFCP $S_t \Rightarrow G_{1:k,t|out}$, pues las características del conjunto de datos implicados en esa clasificación son diferentes. En ese caso, estudiado más adelante, se hace necesario analizar el comportamiento de los algoritmos propuestos de segmentación ante este nuevo tipo de datos.

De los resultados mostrados en el capítulo 5 se concluye que la fiabilidad del “*k-medias secuencial*” depende fundamentalmente del ajuste correcto del parámetro $distM$. Este ajuste ha de realizarse en función del significado del parámetro en la aplicación concreta del clasificador, por lo que en el capítulo anterior no se define un valor concreto para él, si bien se observa que una vez ajustado, el comportamiento del segmentador es robusto a modificaciones de $distM$ en un rango de variación de casi el 30% (ver allí la Figura 5.24). El correcto ajuste de este parámetro es también importante debido a que, como se indica en ese capítulo, $distM$ se comporta como parámetro de gating⁸ del proceso de asociación que el clasificador de medidas $Y_t \Rightarrow G_{1:k,t|in}$ desempeña en el XPFCP, por lo que la creación y eliminación de hipótesis de seguimiento en el tracking depende de él.

En la aplicación del “*k-medias secuencial*” como clasificador del set Y_t en el XPFCP $distM$ especifica el radio de cada clase de medidas $G_{j,t|in} / j = 1:k_{in,t}$, definidas éstas, como ya se ha comentado, a través de un círculo en el espacio de clasificación XZ que incluye a sus miembros $L_{j,t|in}$. El ajuste de $distM$ ha de fijarse, por tanto, en función del tamaño de los objetos implicados en la tarea de MTT de interés, de los que se obtiene el set de medidas Y_t a clasificar, así como del sistema de observación empleado. Por todo ello se emplea la varianza

⁸ El gating se usa en la mayor parte de algoritmos de seguimiento, al igual que en el XPFCP, para crear y eliminar hipótesis de tracking, como se explica en el capítulo 2.

del modelo de ruido de medida O , analizada en un punto anterior, para ajustar el valor de este parámetro. Éste se fija finalmente a $distM_{opt} = \sqrt{(3 \cdot \sigma_{o,x})^2 + (3 \cdot \sigma_{o,z})^2} \approx 640mm$ ⁹, de modo que el radio $distM$ de cada clase $G_{j,t|in}$ incluya un 99.9% de las posibles realizaciones del set de medidas $L_{j,t|in}$ asociadas a ella, en las dos coordenadas de su espacio de definición XZ .

Además de $distM$, el “*k-medias secuencial*” incluye otros siete parámetros asociados con su etapa de validación. De todos ellos el más importante es β_{valid} , pues, como se explica en el apartado 5.2.5 del capítulo 5, define el tamaño de la ventana de validación de cada clase. En el apartado 5.2.6 se muestra que la desviación típica de la tasa de error generada por el clasificador de medidas para un rango de β_{valid} de 1 a 5 es de tan solo el 1.5%, lo que indica la baja sensibilidad del segmentador respecto al valor de este parámetro.

La Figura 6.27, muestra la tasa de error generada por el XPFCP en el experimento complejo, en función del valor asignado a β_{valid} , en un rango de 0 a 5. En esta figura se observa que la desviación típica del error producido por el filtro bayesiano es incluso menor (del 1%) que la obtenida de los resultados generados por el clasificador de medidas, en las mismas condiciones de prueba ($\beta_{valid} = [1,5]$).

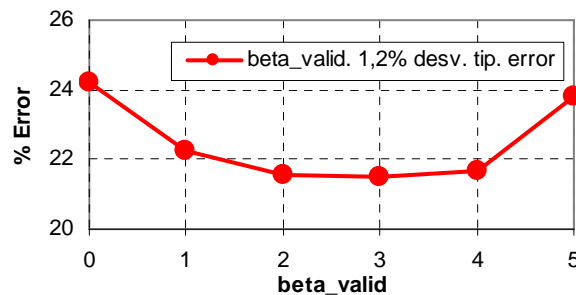


Figura 6.27. Tasa de error obtenida a la salida del XPFCP en función del valor asignado al parámetro β_{valid} (β_{valid}), en el experimento complejo.

Además la Figura 6.27, muestra que la tasa de error del XPFCP con $\beta_{valid} = 0$ es 2.6 puntos superior al 21.6% obtenido con el ajuste óptimo de los parámetros. Este resultado se debe a que la anulación del parámetro β_{valid} conlleva la deshabilitación del proceso de validación del clustering de medidas, hecho que produce un aumento de la tasa de error general del clasificador (ver en el capítulo 5), que se refleja en la fiabilidad del XPFCP a través de la tarea de asociación que éste desempeña en el filtro bayesiano. La conclusión de estos análisis es que el valor de $\beta_{valid,opt}$ ha de fijarse en el rango de 2 a 4 para obtener máxima fiabilidad a la salida del XPFCP (ver en la figura), y que, en cualquier caso, este parámetro no debe anularse.

Por otro lado, el efecto más importante del proceso de validación en la clasificación de medidas es el robustecimiento del algoritmo frente al ruido (la validación aumenta la tasa de rechazo al ruido del “*k-medias secuencial*” del 29.3% al 97.9%). Teniendo en cuenta la tarea de

⁹ Este valor se corresponde con la distancia euclídea, en el espacio XZ , entre dos puntos separados en cada eje $3 \cdot \sigma_{o,i}$ (mm), usando como valor de $\sigma_{o,i} = 150mm$ el de ajuste establecido en un apartado anterior.

este segmentador en el paso de re-inicialización del XPFCP, la habilitación de su etapa de validación es fundamental en el robustecimiento del filtro bayesiano frente el ruido. La Tabla 6.5 muestra el efecto del proceso de validación en la tasa de rechazo al ruido del XPFCP.

Tabla 6.5. Tasa de rechazo al ruido del XPFCP, en el experimento de ruido, en función de la habilitación del proceso de validación del clasificador de medidas incluido en el filtro.

Valor de β_{valid}	0 (sin validación en cluster de medidas)	2 (con validación en cluster de medidas)
% rechazo al ruido	31.7	97.4

El ajuste de los seis parámetros restantes (al margen de β_{valid}) implicados en el proceso de validación de las clases de medidas $Y_t \Rightarrow G_{1:k,t|in}$ tiene una repercusión secundaria en la fiabilidad del XPFCP, por lo que se analiza de forma resumida en los siguientes párrafos, y a partir del estudio de sensibilidad incluido en apartado 5.2.6 del capítulo 5 en relación a ellos:

- El parámetro $k_{valid,opt}$ ha de fijarse suficientemente alto como para que la relación entre p_{valid} y k_t (mostrada en la Figura 5.15) sea lineal para todas las situaciones de clasificación de medidas. Se fija a un valor de 10, suficientemente alto para la aplicación de interés.
- El parámetro d_{valid} limita el error máximo (como distancia euclídea en el espacio XZ en que se definen las clases de medidas) en el valor predicho de la posición de los centroides, y por ende de los objetos asociados a estos. El valor de este parámetro se define, por tanto, a través del modelo de ruido de estado \bar{v}_t , analizado en un punto anterior, del modelo CV usado, pues como se explica en el capítulo 5, es la modificación \widehat{CV} de éste modelo la empleada en el referido proceso de predicción de centroides. El valor del parámetro se fija a $d_{valid,opt} = 425mm^{10}$, de modo que el limite del error esperado incluya un 99.9% de sus posibles realizaciones en las dos coordenadas de definición de las clases de medidas, XZ .
- Los parámetros normalizados de histéresis en la validación de las clases por probabilidad π_p y por distancia π_d se ajustan a su valor medio ($\pi_{p,opt} = \pi_{d,opt} = 0.5$) de modo que los correspondientes procesos de histéresis se lleven a cabo sin alterar mucho la validación por la variable correspondiente. Además el análisis de la sensibilidad incluido en el capítulo 5 indica que la modificación de estos parámetros no tiene un efecto muy importante en la fiabilidad de la clasificación de medidas (la modificación de 0 a 1 de π_p y π_d produce una desviación típica en la tasa de error del clasificador de medidas del 1.9% y del 1.6% respectivamente, en la Figura 5.25).
- Finalmente, el ajuste de los dos parámetros normalizados de validación por probabilidad $p_{valid|max}$ y α_{valid} requiere un análisis especial debido a dos razones: el clasificador de

¹⁰ Siguiendo el planteamiento expuesto en un párrafo anterior para el ajuste del parámetro $distM$, este valor se corresponde con la distancia euclídea en el espacio XZ , entre dos puntos separados en cada eje del espacio $3 \cdot \sigma_{v,i}$ (mm), usando como valor de $\sigma_{v,i} = 100mm$ el de ajuste establecido.

medidas es más sensible a su valor que al de otros parámetros como muestra la misma Figura 5.25 (se produce un 5.6% y 4.1% de desviación típica en la tasa de error del clasificador ante una variación de 0 a 1 de $p_{valid|max}$ y α_{valid} respectivamente); además, ambos parámetros están indirectamente implicados en el proceso de ponderación de las partículas añadido al paso de corrección del XPFCP como se explica en el apartado 6.1.1. La Figura 6.28 muestra la fiabilidad de los resultados generados por el filtro bayesiano, en función de la modificación del valor de ajuste de los dos parámetros comentados: $p_{valid|max}$ ¹¹ y α_{valid} . Los datos presentados en esta figura muestran una sensibilidad media del XPFCP al correcto ajuste de estos parámetros, como era de esperar. El comportamiento del filtro, sin embargo es bastante robusto para modificaciones de hasta el 50% de estos parámetros respecto a su valor de ajuste óptimo $p_{valid|max,opt} = \alpha_{valid,opt} = 0.4$.

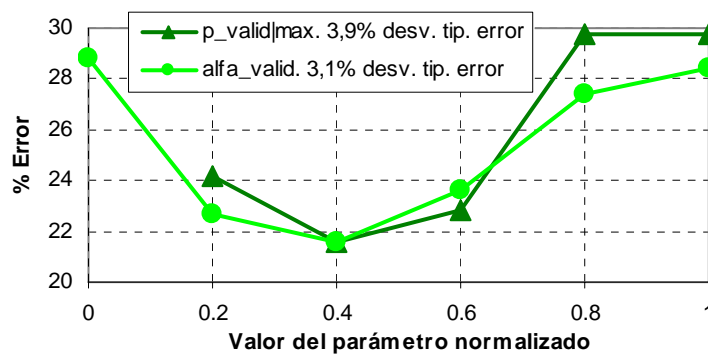


Figura 6.28. Tasa de error obtenida a la salida del XPFCP en función del valor asignado a los parámetros $p_{valid|max}$ ($p_valid|max$) y α_{valid} ($alfa_valid$), en el experimento complejo.

Los análisis expuestos en los párrafos anteriores confirman la robustez del proceso de estimación propuesto en esta tesis a la modificación de los parámetros del clasificador de medidas que incluye. Se completa, por tanto de este modo el análisis de fiabilidad del XPFCP en función de los distintos procesos y parámetros que incluye para mejorar su comportamiento.

6.1.4. Resultados comparativos del XPFCP frente al XPF

En este apartado se presenta una comparativa cuantitativa de la eficiencia que presenta el XPFCP propuesto y ajustado frente a la del XPF (presentado en [Koller-Meier01]). Se incluye esta comparativa para demostrar las mejoras funcionales que aporta la propuesta de estimación multimodal para aplicaciones de MTT desarrollada en esta tesis al algoritmo del que surge, gracias a la incorporación del proceso de clasificación de medidas $Y_t \Rightarrow G_{1:k,t-1|in}$.

Para analizar la mejora funcional comentada, en este punto se emplean los factores de calidad comentados anteriormente: la fiabilidad, la robustez y sensibilidad, y el t_{exe} en la aplicación del filtro propuesto al seguimiento de múltiples objetos en entornos complejos.

¹¹ No se realiza la prueba correspondiente al ajuste de $p_{valid|max} = 0$, pues produce la degeneración del set debido al pesado idéntico de todas las partículas (ver ecuación <6.11>).

a) Comparación de funcionalidad, fiabilidad y robustez frente al ruido

En este punto se incluye un análisis de fiabilidad de la propuesta de estimación multimodal presentada en esta tesis (el XPFCP) frente a la del algoritmo XPF empleado como punto de partida del anterior. Para demostrar la necesidad de incluir en el XPF todas las mejoras expuestas en el punto 6.1.1 (como allí se explica, en los pasos de re-inicialización y corrección de éste), en el estudio comparativo de fiabilidad expuesto se han incluido, además de los resultados generados por los algoritmos comentados (XPF y XPFCP), los arrojados por un algoritmo idéntico al XPF en todas las etapas excepto en la de re-inicialización, en la que incorpora la funcionalidad del XPFCP.

De este modo, en la Figura 6.29 se muestra, en distintos experimentos de MTT, la tasa de error generada por el XPF (en azul intenso), por ese mismo algoritmo con las modificaciones del paso de re-inicialización planteadas en esta tesis (en azul claro), y por el anterior con las propuestas aquí presentadas para el paso de corrección, es decir, el XPFCP (en azul oscuro).

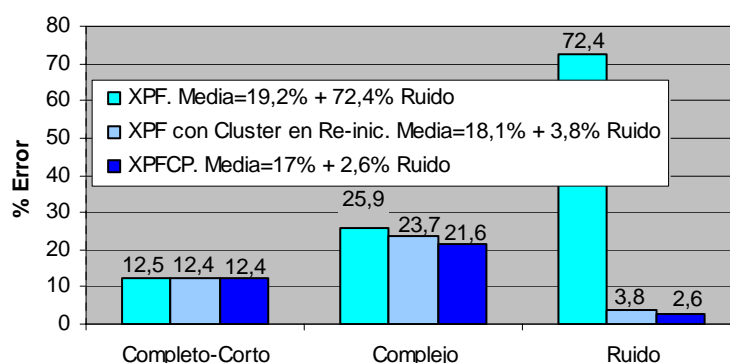


Figura 6.29. Tasa de error generada por el XPF, el XPF con las modificaciones del paso de re-inicialización propuestas en la tesis, y el XPFCP, en los experimentos complejo, completo corto y de ruido.

De la Figura 6.29 se extrae la conclusión de que la fiabilidad del XPF en aplicaciones de seguimiento es similar a la del XPFCP propuesto en experimentos no muy complejos, pero es bastante menor (4.3 puntos menos de tasa de error) que la de la propuesta presentada en esta tesis en situaciones complejas. Este resultado se debe a la ya comentada baja robustez de la representación que tienen las hipótesis nuevas o pobremente sensadas (habituales en contextos complejos de estimación) en la creencia discreta generada a la salida del XPF, y que es reforzada gracias a las modificaciones de los pasos de re-inicialización (incluyendo estas modificaciones en la funcionalidad del XPF su tasa de error disminuye un 2%) y corrección, que dan lugar al XPFCP.

La Figura 6.29 muestra también la mejora que produce la incorporación del proceso de clasificación de medidas $Y_{t-1} \Rightarrow G_{1:k,t-1|in}$ a los pasos de re-inicialización y corrección del flujograma del XPF, en el comportamiento del algoritmo final frente a los outliers: el filtrado del ruido que, entre otras tareas, lleva a cabo este clasificador, aumenta la tasa de rechazo al ruido del XPF en casi un 70%. La figura permite concluir además, que la aplicación de la salida del segmentador a la etapa de re-inicialización del XPF es fundamental desde el punto de vista

del filtrado de outliers, pues la tasa de rechazo al ruido de este algoritmo así modificado es muy semejante a la del XPFCP.

Se comprueba así una de las ventajas ya comentadas del uso de la salida del clasificador de medidas $Y_{t-1} \Rightarrow G_{1:k,t-1|in}$ como base del proceso de incorporación de partículas $S_{t-1} \Rightarrow \widehat{S}_{t-1}$ en el paso de re-inicialización del estimador multimodal: el alto rechazo al ruido del segmentador (97.9%, según se analiza en el capítulo 5) implica una baja probabilidad de que los outliers se incorporen como hipótesis nuevas al set de salida \widehat{S}_{t-1} de este paso de re-inicialización, y por ende, a la creencia discreta $p(\bar{x}_t^{(1:n-n_{m,t})} | \bar{y}_{1:t})$ que da como resultado de estimación el XPFCP.

La Figura 6.30 ejemplifica, por otro lado, la funcionalidad de las modificaciones propuestas para el paso de corrección en el XPFCP, en la aplicación de MTT de interés. La figura incluye dos imágenes (organizadas en columnas) extraídas de la ejecución en tiempo real de las tres versiones del algoritmo analizadas en este apartado: el XPF (columna izquierda), el mismo con las modificaciones del paso de re-inicialización propuestas en esta tesis (columna central), y el XPFCP (columna derecha). La fila de imágenes superiores muestra la representación discreta de la PDF a la salida del paso de corrección $S'_t = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ de cada algoritmo; y la fila inferior muestra la del set de partículas S_t a su salida (puntos rojos) y de medidas Y_t (puntos verdes) en la imagen de entrada a la tarea de seguimiento.

Como se observa en la PDF discreta caracterizada por el set $S'_t = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ a la salida del paso de corrección de los tres algoritmos analizados (fila superior de imágenes de la Figura 6.30), las distintas hipótesis de seguimiento existentes en dicho instante temporal (cuatro personas en primer plano, en las imágenes de la fila inferior de la figura) están representadas en cualquier caso. Sin embargo su representación individual en distintos modos de esta PDF es sólo apreciable en la salida que genera el XPFCP (imagen superior derecha de la figura). Las propuestas de robustecimiento de las distintas hipótesis de seguimiento en función de la clase de medidas $Y_t \Rightarrow G_{1:k,t|in}$ con la que cada una de ellas se relaciona, incluidas en la etapa de corrección del XPFCP permiten, por tanto, que la creencia discreta a la salida de este filtro presente tantos modos como objetos hay implicados en la tarea de estimación en cada instante. Este robustecimiento individual facilita, además, la posterior segmentación del set de salida que representa a la creencia $S_t \cong p(\bar{x}_t^{(1:n-n_{m,t})} | \bar{y}_{1:t})$ con un proceso de clasificación de baja carga computacional, como el analizado en el apartado siguiente, difícilmente realizable sobre el set S_t generado por el XPF sin o con las modificaciones del paso de re-inicialización planteadas en la tesis (imágenes superiores izquierda y central de la figura).

El peor comportamiento del XPF (sin o con las modificaciones del paso de re-inicialización comentadas) en situaciones complejas de seguimiento, mostrado en la Figura 6.29, se debe, por tanto, a que el set de partículas S_t a su salida no representa específicamente las distintas hipótesis de seguimiento, como se observa en esta última Figura 6.30.

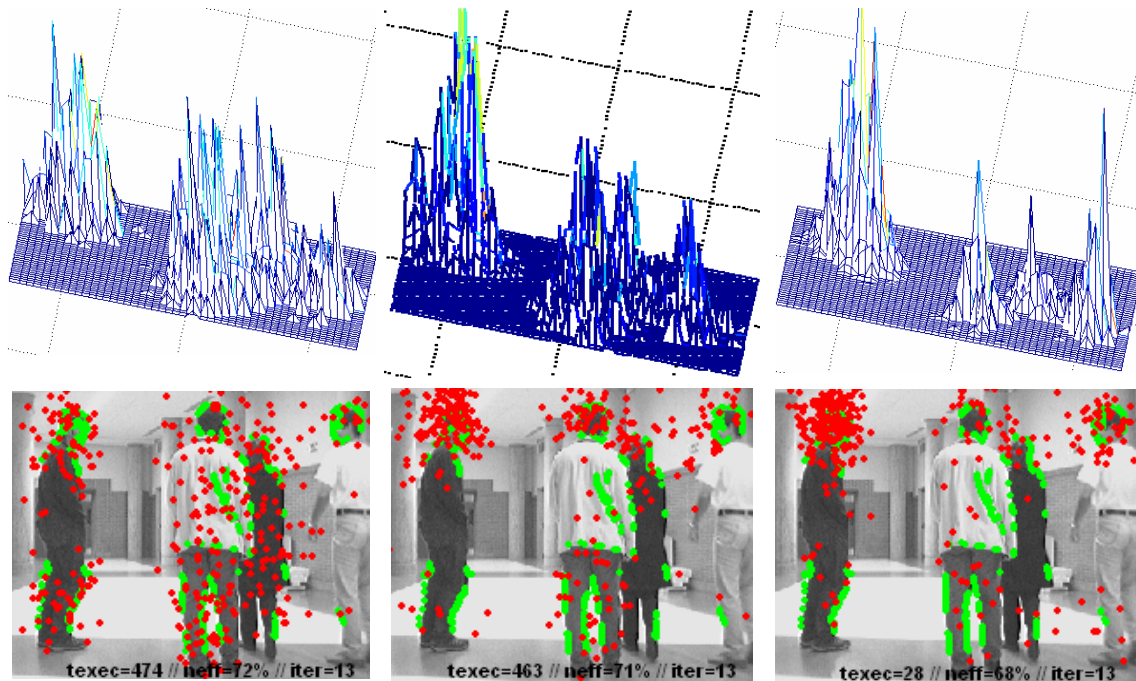


Figura 6.30. Imágenes extraídas de la ejecución en el experimento complejo en tiempo real del XPF (columna izquierda de imágenes), el XPF con las modificaciones del paso de re-inicialización propuestas en la tesis, (columna central) y el XPFCP (columna derecha). La fila superior de imágenes es una representación discreta de la creencia representada por el set $S'_t = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ a la salida del paso de corrección, y la fila inferior la representación del set S_t de salida de los filtros en la imagen de entrada a éstos. En estas últimas gráficas los puntos rojos representan la proyección de la posición caracterizada por cada partícula de S_t en el espacio UV, y los verdes de las medidas de Y_t en el mismo espacio. Además se incluye el t_{exe} y el valor de \hat{n}_{eff} (neff) generado por cada uno de los algoritmos en el frame mostrado.

El análisis anterior permite además concluir, que el incremento del número de partículas n total en el XPF o del número $n_{m,t}$ de éstas que se incorpora al set S_{t-1} en el paso de re-inicialización no mejora el comportamiento de este filtro en situaciones complejas de estimación. Esta conclusión permite descartar el estudio de fiabilidad del XPF frente a los dos parámetros mencionados, pues el análisis anterior permite anticipar que el ajuste de éstos no mejorará la fiabilidad del XPF en situaciones complejas.

Por otro lado, la Figura 6.31 muestra el valor de \hat{n}_{eff} generado por el XPF, este algoritmo con las modificaciones propuestas para el paso de re-inicialización, y el XPFCP, en el experimento complejo. Se incluye también en la figura una tabla que recoge el valor medio de \hat{n}_{eff} en el mismo experimento ((a), mostrado también en trazo grueso del mismo color en la gráfica de la figura) y en el completo corto (b).

La Figura 6.31 muestra que, desde el punto de vista del factor de calidad \hat{n}_{eff} , el XPF es algo más robusto que el XPFCP aquí propuesto, si bien en todos los casos el valor de \hat{n}_{eff} es superior al límite del 66% normalmente establecido para asegurar la eliminación del problema de empobrecimiento característico del PF. Este resultado se corresponde con la correcta

reproducción del conjunto global de partículas en el paso de resampling de este algoritmo, y no con la robustez de las distintas hipótesis caracterizadas por el set que se obtiene a su salida S_t , como muestra la Figura 6.30.

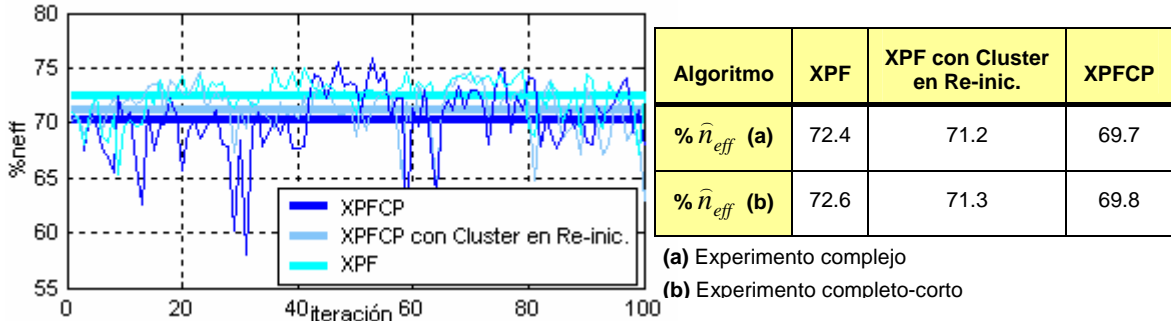


Figura 6.31. Gráficas comparativas del valor de \hat{n}_{eff} generado en el experimento complejo por el XPF, el XPF con las modificaciones al paso de re-inicialización propuestas, y el XPFCP. Las líneas gruesas muestran el valor medio de las de trazo fino. Se incluye una tabla con el valor medio de \hat{n}_{eff} en ese test (a) y en el completo corto (b).

Para analizar mejor la robustez con la que los algoritmos estudiados representan en su set de salida S_t las hipótesis de seguimiento más débiles (por su novedad o a la pobreza del conjunto de medidas que las representan en el set de entrada Y_t), se obtiene la gráfica mostrada en la Figura 6.32 en el experimento complejo, donde aparecen con más frecuencia situaciones con hipótesis débiles de seguimiento. En esta figura se representa el número de “hijos” obtenido tras el paso de selección por cada una de las partículas insertadas en el de re-inicialización $\hat{s}_{i,t-1} / i = n - n_{m,t-1} : n$ (relativo al número de partículas $n_{m,t-1}$ insertadas en S_{t-1} en esa etapa), pues son éstas y no el resto, las partículas que refuerzan la representación de estas hipótesis débiles a la salida de los estimadores.

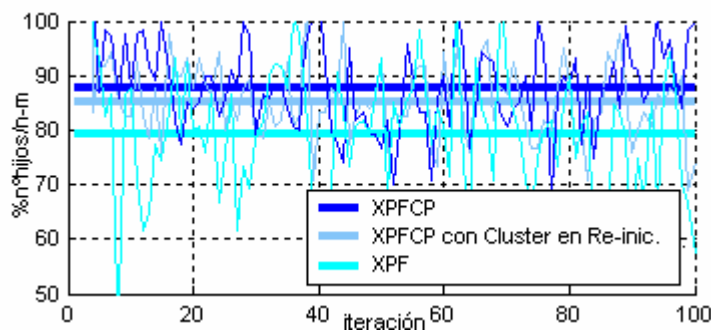


Figura 6.32. Gráficas comparativas del número de “hijos” obtenido para cada partícula insertada en el paso de re-inicialización $\hat{s}_{i,t-1} / i = n - n_{m,t-1} : n$ relativo al número de partículas $n_{m,t-1}$ insertadas en S_{t-1} en esa etapa, para los mismos algoritmos y en las mismas condiciones de la Figura 6.31.

De la Figura 6.32 se extrae que la tasa de reproducción en el resampling de las partículas insertadas en el paso de re-inicialización $\hat{s}_{i,t-1} / i = n - n_{m,t-1} : n$ es, de media un 8.5% superior en el XPFCP que en el XPF, en situaciones complejas de estimación, superando además en más

de un 7.5% el mejor valor esperado para esta tasa¹²: la propuesta de re-inicialización planteada mejora esta tasa media de reproducción generada por el XPF (del 79.2%) en un 6% (alcanzando un 85.2%) y la de corrección en un 1.5% más (llegando al 87.6%).

Los análisis expuestos permiten concluir que las modificaciones propuestas para los pasos de re-inicialización y corrección del XPFCP aportan robustez a la representación de las hipótesis más débiles en la creencia obtenida a la salida del filtro bayesiano, y por ende, mejoran la fiabilidad del XPF propuesto en [Koller-Meier01] insuficiente, según se comenta a lo largo de esta memoria, para llevar a cabo con éxito la tarea de MTT de interés en la tesis.

b) Comparación de tiempo de ejecución

Otra característica del XPFCP que lo hace más adecuado que el XPF para la aplicación de MTT de interés es su menor coste computacional. Como se expone en puntos anteriores, el clasificador de medidas $G_{1:k_t|in}$ que se añade al XPF para dar lugar al XPFCP comprime la información en el set $Y_t = \left\{ \bar{y}_t^{(i)} \right\}_{i=1}^{m_t}$. El uso de la información contenida en las $k_{in,t}$ clases en los pasos de re-inicialización y corrección del filtro, en lugar de la encerrada en las m_t medidas disminuye bastante el t_{exe} del XPFCP: este tiempo pasa de depender del número de medidas m_t a hacerlo del número mucho menor de clases de éstas $k_{in,t}$.

La Tabla 6.6 muestra el valor medio del t_{exe} de los dos algoritmos en el experimento complejo. Los resultados presentados en esta tabla demuestran los comentarios del párrafo anterior: el t_{exe} del XPF es aproximadamente 25 veces superior al del XPFCP en situaciones normales y 30 en el experimento complejo. De hecho, la mayor parte del t_{exe} del XPF (el 99.7% de éste en el experimento complejo) se consume en su paso de corrección, que tiene una complejidad computacional exponencialmente dependiente del número de medidas de entrada al filtro $O(n^{m_t})$, mientras que esta etapa consume sólo el 22% del t_{exe} del XPFCP en las mismas condiciones de prueba, pues la complejidad computacional del filtro propuesto es idéntica pero respecto al número mucho menor de clases de medidas $O(n^{k_{in,t}})$.

Tabla 6.6. Valor medio del t_{exe} del XPFCP y del XPF al ejecutarlos en los experimentos complejo y completo corto.

Test/Algoritmo	XPF	XPFCP
t_{exe} (s) Complejo	0.826	0.027
t_{exe} (s) Completo-Corto	0.493	0.020

¹² La tasa de reproducción media esperada del set de partículas en el paso de resampling depende de la diferencia entre el número de éstas de entrada y de salida de esta etapa: $\frac{n - n_{m,t-1}}{n} = 1 - \gamma_t$. En el caso del XPFCP esta relación la determina, por tanto, el valor de γ_t , que en los experimentos mostrados está en el intervalo del 20% al 30%, a partir de las conclusiones obtenidas en apartados anteriores. El valor medio esperado de esta tasa de reproducción está, por tanto en el entorno del 70% al 80%.

El algoritmo propuesto en la tesis se ejecuta, por tanto, en tiempo real como se especifica en el capítulo 1, bajo todas las condiciones de test (como demuestra la Tabla 6.6), mientras que el XPF no lo hace. Conviene recordar que el incumplimiento de la especificación de tiempo real (sobre todo en entornos muy poblados como los aquí analizados) es un problema habitual de muchas de las propuestas de MTT planteadas por la comunidad científica, como se comenta en el capítulo 2. El XPFCP propuesto en esta tesis alcanza esta especificación incluso en entornos complejos, mejorando así el comportamiento de otros algoritmos como se demuestra en la comparativa, incluida en el capítulo siguiente, del XPFCP con dos de estas propuestas.

Queda empíricamente comprobado, por tanto, que el XPFCP mejora el comportamiento del algoritmo en que se basa, el XPF propuesto en [Koller-Meier01], en lo que se refiere a fiabilidad, robustez y t_{exe} . El XPFCP alcanza las especificaciones establecidas en el capítulo 1 para su aplicación a la tarea de MTT de interés, que no cumple el XPF.

6.1.5. Conclusiones sobre los resultados expuestos

Como comentario final sobre todos los resultados expuestos en los dos apartados anteriores, se concluye que el XPFCP diseñado en la tesis como propuesta de estimación multimodal para la aplicación de MTT de interés, cumple con las especificaciones impuestas en el capítulo 1:

1. Su tasa de fiabilidad alcanza una media del 82.7% en los experimentos que por su complejidad se emplean en las distintas pruebas aquí mostradas (ver apéndice III). Este valor es mayor tras el proceso de clasificación de partículas $S_t \Rightarrow G_{1:k,t|out}$, descrito en el apartado siguiente, elevándose por encima del 90% si sólo se tienen en cuenta situaciones mantenidas de error, como se muestra al final de este capítulo.
2. El t_{exe} del algoritmo en la plataforma de pruebas, permite asegurar la ejecución del seguidor multi-objeto a una velocidad próxima a los 40FPS, superior a la de la mayor parte de soluciones de MTT propuestas por la comunidad científica (ver en el capítulo 2).
3. Las mejoras incluidas en la funcionalidad del XPFCP aseguran la eliminación del problema de empobrecimiento y degeneración del set característico de los estimadores basados en el PF. Afirmación justificada con la obtención de valores del factor de calidad \hat{n}_{eff} superiores ($\hat{n}_{eff} = 69.8\% > 66\%$) al mínimo establecido por la comunidad científica.
4. Finalmente, la flexibilidad del estimador multimodal propuesto queda demostrada en su baja sensibilidad a la modificación de varios de sus parámetros, fundamentalmente relacionados con el número de partículas empleadas para su correcta ejecución (n , n_m y n_{init}) y con el algoritmo de clasificación incluido (descrito en el capítulo 5). Por su parte, el proceso de clasificación de medidas $Y_t \Rightarrow G_{1:k,t|in}$ actúa como filtro, insensibilizando la eficiencia de estimación del XPFCP frente a los parámetros del modelo de observación, y más concretamente, frente al ruido \bar{o}_t incluido en el set de medidas Y_t , entrada del estimador bayesiano. Este hecho facilita su aplicación en tiempo real a tareas de MTT basadas en distintos métodos de sensado, como se demuestra en un apartado posterior.

5. Además, los resultados comparativos obtenidos del comportamiento del XPFCP frente al del XPF en el que aquél se basa, indican que el primero supera al segundo en todos los aspectos de interés en su aplicación a la tarea de seguimiento (fiabilidad ante hipótesis pobres, robustez frente al ruido y a la modificación de los parámetros y t_{exe}). El objetivo de modificar el XPF para poder ser usado con éxito en tareas de MTT, planteado al principio de este capítulo, se alcanza con las modificaciones propuestas en esta tesis, que dan lugar al XPFCP, tal y como demuestran los análisis teóricos y empíricos aquí incluidos.

Los resultados mostrados en el apartado siguiente completan las conclusiones aquí expuestas sobre la idoneidad del proceso probabilístico propuesto para la tarea de seguimiento de múltiples objetos en entornos complejos de interés.

6.2. Incorporación del proceso de clasificación a la salida del XPFCP

Como se expone en el capítulo 3, a la salida del XPFCP, y fuera de su lazo de ejecución recursiva, se incorpora un nuevo proceso de clasificación, cuya funcionalidad es proporcionar una solución determinística a partir de la creencia discreta $p(\bar{x}_t^{(n-n_{m,t})} | \bar{y}_{1:t})$ que genera el filtro.

La clasificación en esta etapa del algoritmo se hace, por tanto, sobre el conjunto de partículas $S_t = \left\{ \bar{x}_t^{(i)}, \frac{1}{n-n_{m,t}} \right\}_{i=1}^{n-n_{m,t}} \Rightarrow G_{1:k,t|out}$ que genera el XPFCP a la salida del paso de selección (ver zona sombreada en verde en la Figura 6.1) y es, por tanto, diferente al que se usa para robustecer el funcionamiento del estimador multimodal, descrito en los apartados anteriores, pues aquel se ejecuta sobre el conjunto de medidas $Y_t \Rightarrow G_{1:k,t|in}$ de entrada al filtro bayesiano.

Las características de los datos a clasificar (el set de partículas S_t en el caso del segmentador estudiado en este apartado y el de medidas Y_t analizado en el 6.1) son también distintas: el número total de datos a organizar (n para el segmentado de partículas, y m_t para el de medidas) y su distribución entre las clases es más constante y equilibrada en este caso ($S_t \Rightarrow G_{1:k,t|out}$) que en el anterior $Y_t \Rightarrow G_{1:k,t|in}$. El clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$ tiene, por ello, especificaciones diferentes a las del de medidas $Y_t \Rightarrow G_{1:k,t|in}$. En el capítulo 5 se analiza la funcionalidad de los algoritmos propuestos en función de estas especificaciones, concluyéndose que el clasificador borroso parece más adecuado para la clasificación de sets de datos de distribución homogénea que el booleano. Debido a ello, el “*subtractive enlazado*” con identificación y validación se propone como solución más adecuada al problema de clasificación de partículas aquí descrito.

En cualquier caso, en este apartado se analiza teórica y empíricamente, del mismo modo que se hizo con el segmentador de medidas $Y_t \Rightarrow G_{1:k,t|in}$ en el caso anterior, la funcionalidad y eficiencia de las dos propuestas de clasificación en la tarea de segmentación de partículas

$S_t \Rightarrow G_{1:k_t|out}$. El análisis incluye también en este caso un estudio de fiabilidad, t_{exe} robustez y sensibilidad frente a sus parámetros.

6.2.1. Descripción general de la clasificación de partículas a la salida del XPFCP

La salida del clustering de partículas es un conjunto de clases de partículas $G_{j,t|out} / j=1:k_{out,t}$ que representan a los distintos picos de hipótesis caracterizados en la creencia discreta a la salida del algoritmo XPFCP $p(\bar{x}_t^{(1:n-n_{m,t})} | \bar{y}_{1:t})$. La aplicación del proceso de estimación multimodal desarrollado por este algoritmo a la tarea de MTT de interés, junto a la clasificación incluida a la salida de éste, permite generar el conjunto de grupos $G_{j,t|out} / j=1:k_{out,t}$, que caracterizan a la solución de tracking del conjunto de objetos presentes en t en el entorno de seguimiento.

Gracias al uso de los algoritmos de clasificación propuestos en la tesis en la tarea de clasificación de partículas aquí analizada cada hipótesis de seguimiento obtenida $G_{j,t|out} / j=1:k_{out,t}$ incluye las características descritas en el capítulo 5 (ecuación <5.44>), incluyendo todas las planteadas como objetivo en el capítulo 1 de la tesis:

- *Posición y velocidad estimada del centroide del objeto:*

$$\bar{g}_{j,t|out} = [x_{j,t|out} \quad y_{j,t|out} \quad z_{j,t|out} \quad vx_{j,t|out} \quad vz_{j,t|out}] / j=1:k_{t|out}$$
- *Identidad del objeto:* $\tau_{j|out} / j=1:k_{t|out}$
- *Probabilidad de la estimación:* $p_{j,1:t|out} / j=1:k_{t|out}$; y *validez:* $\beta_{j,t|out} / j=1:k_{t|out}$

Por otro lado, como se comenta en párrafos anteriores, la aplicación de los algoritmos de clasificación propuestos para la organización del conjunto de partículas $S_t \Rightarrow G_{1:k_t|out}$ presenta ciertas diferencias frente a la que se realiza sobre el conjunto de medidas $Y_t \Rightarrow G_{1:k_t|in}$, analizada en el apartado anterior. Estas se analizan en detalle a continuación:

1. Las componentes del vector de datos a clasificar cambian, pasando de ser $\bar{y}_t = [x_t \quad y_t \quad z_t]^T$, en el caso de la clasificación del set de medidas Y_t , a ser $\bar{x}_t = [x_t \quad y_t \quad z_t \quad vx_t \quad vz_t]^T$ en la clasificación del de partículas S_t , según las ecuaciones del modelo CV usado en la estimación (<2.6> y <2.8> del capítulo 2).

Debido a este cambio, el espacio de clasificación usado pasa de ser bidimensional (XZ) a definirse en cuatro dimensiones ($XZvXvZ$), pues la clasificación se hace en ambos casos sin considerar la componente de altura y , debido a su nula aportación de información discriminante en los dos procesos de clasificación (ver en el capítulo 5).

La diferencia en este punto entre los clasificadores radica, por tanto, en que las componentes de velocidad en el espacio XZ ($[vx \quad vz]^T$) de las hipótesis de tracking caracterizadas por el set S_t se incluyen también en el proceso de clasificación de partículas $S_t \Rightarrow G_{1:k,t|out}$. Este hecho tiene como efecto positivo el robustecimiento de la clasificación debido al aumento del número de variables usadas, y como negativo el aumento del t_{exe} del clasificador.

2. Debido a la diferencia comentada en el párrafo anterior, el proceso de predicción de centroides $\bar{g}_{1:k,t|out}$, base de la identificación de las clases en los clasificadores propuestos, se realiza a partir del propio modelo CV, evitando el uso de su versión modificada (\widehat{CV}) expuesta en el apartado 5.2.4. del capítulo 5. Este proceso es, por tanto, independiente (tanto en el “*k-medias secuencial*” como en el “*subtractive enlazado*” con identificación) del de identificación o asociación entre cada clase $G_{j,t|out}$ y el objeto $j=1:k_{out,t}$ al que ésta representa, pues su vector de entrada (velocidad del centroide $\vec{u}_{j,t|out}$, dada por la ecuación <5.17>) se obtiene del propio proceso de clasificación al estar incluido en el vector de datos a clasificar ($\vec{x}_t = [x_t \quad y_t \quad z_t \quad vx_t \quad vz_t]^T$ en la clasificación de partículas, en lugar de $\vec{y}_t = [x_t \quad y_t \quad z_t]^T$ usado en la de medidas).
3. El valor de ajuste de los parámetros de los clasificadores ha de revisarse respecto al expuesto en el apartado 6.1.3, teniendo en cuenta las ya comentadas distintas características del conjunto de datos a organizar (S_t frente a Y_t). Esta revisión afecta fundamentalmente a los parámetros de verosimilitud del proceso de validación (dada la diferente distribución entre las clases de S_t frente a la de Y_t), y a los que establecen los límites de distancia que definen la generación de una clase nueva, en el espacio de clasificación correspondiente. El análisis de estos ajustes se incluye en un apartado posterior.

El cambio en la definición de las componentes del vector de datos de entrada al clasificador de partículas conlleva una serie de modificaciones en lo referente a la caracterización del objeto al que representa cada una de sus clases de salida $S_t \Rightarrow G_{1:k,t|out}$. Esta caracterización depende del tipo de medida de distancia (definida en el espacio $XZvXvZ$) usada como característica de clasificación del set de partículas S_t . En este punto se propone emplear la distancia euclídea como característica de clasificación, pues el conjunto S_t cumple las mismas especificaciones (planteadas en el apartado 6.1.1.) que establecen el uso de esta forma de distancia en la clasificación del set de medidas $Y_t \Rightarrow G_{1:k,t|in}$.

De este modo, cada grupo de partículas a la salida del clasificador $S_t \Rightarrow G_{1:k,t|out}$ queda representado por un círculo en XZ , cuyo centroide coincide con el de la clase $\bar{g}_{j,t|out} / j=1:k_{out,t}$, que se convierte en un cilindro al prolongarse en la coordenada de altura (y) no incluida en el espacio de clasificación. Cada cilindro tiene además una velocidad dada por la de su centroide $\vec{g}_{j,t|out}$, con un margen de error establecido de nuevo por la distancia euclídea usada en la clasificación, aplicada esta vez al espacio de velocidad $vXvZ$. La velocidad

del objeto queda definida de este modo en el espacio XZ por una flecha indicadora de la intencionalidad de movimiento del objeto asociado a cada cilindro, cuya una magnitud y dirección presenta una incertidumbre circular (al igual que la que define la posición del objeto).

La Figura 6.33 muestra la representación en el espacio XYZ , de los resultados generados por el clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$, y por ende, de la aplicación global de MTT, en un ejemplo real de tracking de dos objetos, $k_{out,t} = 2$. En ella se muestran, en distintos colores, los cilindros indicadores de la posición estimada que ocupa cada objeto seguido. Se dibuja también por cada objeto su velocidad estimada, cuya magnitud (como movimiento del centroide en 1s) y dirección quedan representados con una flecha con origen en el centroide del cilindro $\bar{g}_{1:2,t|out}$ (marcado con un aspa), del mismo color pero en tono más claro que el cilindro correspondiente. En la figura se incluye además el valor numérico de las coordenadas en el espacio de clasificación de partículas ($[x_j \ z_j \ vx_j \ vz_j] / j = 1:2$, donde x_j y z_j se indican en m , y vx_j y vz_j en m/s), generadas para cada centroide por el clasificador $\bar{g}_{1:2,t|out}$. Este vector constituye la estimación final de posición y velocidad generada por el XPFCP para cada objeto.

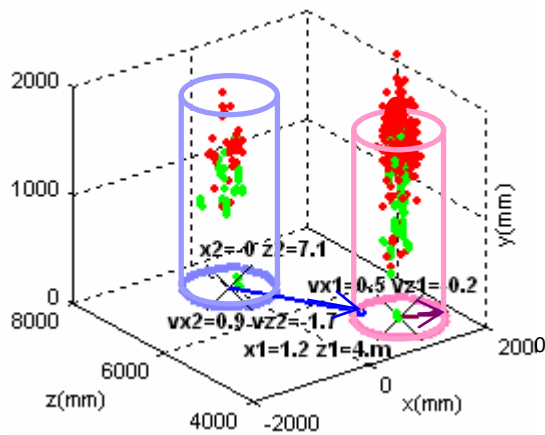


Figura 6.33. Ejemplo de representación en el espacio XYZ del resultado que genera el clasificación de partículas $S_t \Rightarrow G_{1:k,t|out}$, incluyendo con distintos colores el cilindro que caracteriza la estimación de posición de cada objeto relacionado con cada clase de salida (con el valor numérico de su centroide en m en el espacio XZ), y una flecha de color más claro indicadora de la magnitud (como movimiento del centroide en 1s) y dirección de su velocidad (con el valor numérico de ésta en m/s en el espacio $vXvZ$).

La Figura 6.34 muestra la proyección en el espacio XZ (dibujo izquierdo) y en la imagen obtenida con el sistema de sensado usado (dibujo derecho), de los resultados presentados en la Figura 6.33. El cilindro que representa la posición de cada uno de los $k_{out,t} = 2$ objetos seguidos se proyecta como un círculo en el plano XZ (a la derecha) y como un rectángulo en el plano imagen (a la izquierda). Los dibujos muestran también la flecha que caracteriza la velocidad de cada objeto, incluyéndose en la representación en el plano XZ (dibujo izquierdo) otro círculo indicador de la incertidumbre de esta velocidad en un t_s (descrita en el párrafo previo).

En las figuras referidas (Figura 6.33 y Figura 6.34) se muestra también con puntos verdes la proyección correspondiente del conjunto de medidas 3D, Y_t ; y con puntos rojos la de la

posición representada por cada partícula de S_t a la salida del XPFCP. En el apartado siguiente se incluyen los análisis de funcionalidad que validan el uso de las propuestas de clasificación descritas en el capítulo 5 (ya empleadas en segmentación de medidas $Y_t \Rightarrow G_{1:k,t|in}$ que requiere el XPFCP), en la clasificación del set de partículas $S_t \Rightarrow G_{1:k,t|out}$ que genera el filtro bayesiano como salida determinística de la estimación.

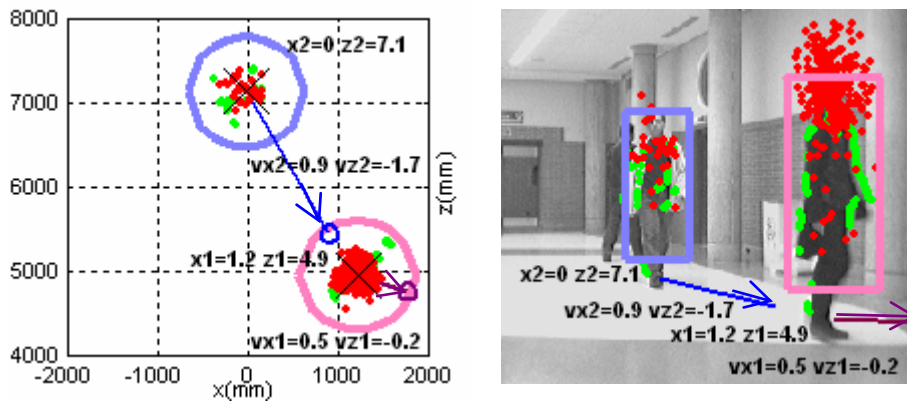


Figura 6.34. Ejemplo de representación en el espacio XZ (dibujo izquierdo) y en la imagen obtenida con el sistema de sensado (dibujo derecho), de los resultados presentados en la Figura 6.33. Además de la proyección de los datos mostrados en la Figura 6.33 sobre estos espacios, la representación en el plano XZ (dibujo izquierdo) incluye sendos círculos indicadores de la incertidumbre de velocidad en un t_s del objeto correspondiente.

6.2.2. Análisis funcional del clasificador de partículas a la salida del XPFCP

En este punto se incluye el análisis funcional de los clasificadores presentados en el capítulo 5, como segmentadores del set de partículas a la salida del XPFCP. Se incluye aquí un estudio de la validez para esta tarea de las dos propuestas allí descritas: el “*k-medias secuencial*” con validación, y el “*subtractive enlazado*” con identificación y validación. Además, se analiza también en este punto, el ajuste de los parámetros de estos clasificadores en su aplicación a la clasificación de partículas.

A lo largo de todo el estudio aquí incluido se hace referencia a la aplicación de los mismos algoritmos a la clasificación de medidas, descrita en apartados previos, contemplando el efecto que tienen en la funcionalidad y el ajuste de sus parámetros, las diferentes características de los datos implicados en cada una de las segmentaciones comentadas en el punto anterior.

El análisis se realiza en base a los mismos factores de calidad empleados en estudios anteriores: la fiabilidad, la robustez y sensibilidad, y el t_{exe} del clasificador de partículas y, por ende, de la salida determinística del XPFCP que ésta define en su aplicación a la tarea de MTT.

Es necesario destacar, finalmente, que la fiabilidad del segmentador vuelve a medirse en este apartado en función de todos los errores tipificados en el apéndice III y empleados ya en los análisis empíricos realizados en el capítulo 5, algunos de los cuales no se usan en los análisis incluidos en los apartados 6.1.3 y 6.1.4 anteriores por carecer de sentido, a saber: error de no

generación de una clase de partículas $G_{j,t|out}$ asociada a algún objeto, de unión de 2 clases en 1, de desplazamiento de alguna de ellas y de duplicación (asignación de 2 clases a un solo objeto).

Es importante recordar, por otro lado, que estos tipos de error no tienen la misma relevancia, como se explica en el apéndice III, en la aplicación de MTT de interés. Además, en muchos de los trabajos analizados en la tesis (como [Schulz03b]) los fallos por no generación de una hipótesis de seguimiento sólo se tienen en consideración cuando se producen durante un mínimo de ejecuciones del tracker. Estas consideraciones son analizadas en el resumen de los resultados obtenidos, incluido al final de este capítulo.

a) Efecto del tipo de algoritmo de clasificación empleado en la fiabilidad de la estimación

Debido a sus características funcionales, analizadas en el capítulo 5, y comentadas en apartados anteriores, el “*subtractive enlazado*” con identificación y validación se plantea más adecuado que el “*k-medias secuencial*” con validación, propuestos en esta tesis, para la clasificación de partículas que, como ya se ha comentado, da lugar a la salida determinística del proceso de estimación probabilístico desarrollado por el XPFCP.

En cualquier caso, en este punto se incluye un análisis empírico de las ventajas de uso de uno u otro algoritmo en la tarea de clasificación de partículas perseguida $S_t \Rightarrow G_{1:k,t|out}$, en lo referente a la fiabilidad de la salida que genera.

La Figura 6.35 muestra la tasa de error obtenida en los experimentos complejo y completo corto de MTT, en función de la propuesta de clasificación de partículas empleada. A la vista de esta figura, se concluye que el “*k-medias secuencial*” y validación tiene una mayor fiabilidad que el “*subtractive enlazado*” con identificación y validación en esta tarea (4 puntos de media).

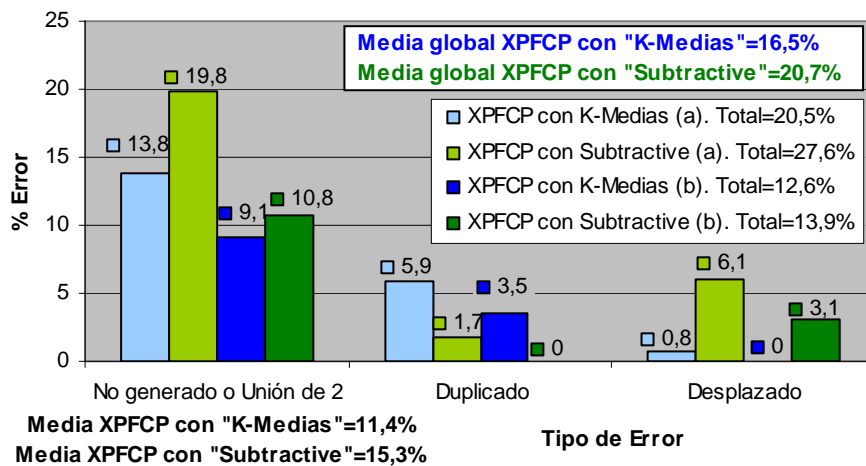


Figura 6.35. Tasa de error generada por el XPFCP en función del algoritmo de clasificación de medidas usado (“*k-medias secuencial*” con validación en azul, o “*subtractive enlazado*” con identificación y validación en verde), en el experimento complejo (tonos claros) y en el completo corto (tonos oscuros).

Esta conclusión es contraria a lo supuesto en apartados anteriores, por lo que es conveniente analizar el resultado en detalle:

- En el capítulo 5 se justifica la mayor idoneidad del clasificador borroso para la organización de partículas S_t en su menor tasa de error por duplicación, cuestión confirmada en la Figura 6.35 (esta tasa es, casi 4 puntos menor en los resultados del “*subtractive enlazado*” con identificación y validación que en los del “*k-medias secuencial*” con validación).
- Sin embargo, la diferencia expuesta en el punto anterior carece de importancia si la tasa de error por no generación de hipótesis del clasificador borroso es mayor que la del booleano, como ocurre en los experimentos aquí analizados (esta tasa de error del borroso es, de media, 4 puntos mayor que la del booleano). Los resultados obtenidos con el “*subtractive*” como segmentador del set de medidas Y_t (ver apartado 6.1.3), muestran esta menor fiabilidad del clasificador borroso frente al booleano, sin embargo, esos análisis justifican la diferencia con el mal comportamiento de este clasificador frente a la irregular distribución del set de datos Y_t . En la clasificación de partículas S_t esta distribución es, como ya se ha comentado, mucho más uniforme, y efectivamente comparando los resultados que generan los clasificadores en la organización del set de medidas Y_t (ver la Figura 5.56 del capítulo 5), y del de partículas S_t , los del algoritmo borroso experimentan una mayor mejora (la tasa de error total se reduce en algo más del 55%) que la que sufren los del booleano (esta tasa se reduce en un 30%), si bien este hecho sólo se produce en el experimento completo corto.
- El comportamiento de ambos algoritmos como clasificadores del set de partículas es, por tanto, equivalente, en situaciones normales de funcionamiento, pues la tasa de error por hipótesis no generadas es similar en estos casos para ambos clasificadores (9.1% del booleano frente a 10.8% del borroso), y la de duplicación de clases del clasificador de tipo “*k-medias*” es similar a la de desplazamientos del borroso (3.5% frente a 3.1%). Sin embargo, el “*k-medias secuencial*” con validación tiene una mejor respuesta como clasificador de partículas en situaciones complejas que el “*subtractive enlazado*” con identificación y validación. Como se explica en el capítulo 5, este hecho se debe a que el proceso de predicción de centroides es intrínseco al algoritmo booleano y no lo es al borroso, por lo que, en situaciones complejas de clasificación, los resultados generados por el primero son mejores que los producidos por el segundo.

La Figura 6.36 muestra un ejemplo de la diferente fiabilidad de los algoritmos propuestos para la clasificación del set de partículas S_t en una situación real extraída del experimento complejo, en el que es más evidente esta diferencia. En la figura se muestran la segmentación realizada por cada una de las dos propuestas planteadas (booleana a la izquierda y borrosa a la derecha), caracterizada del modo mostrado en la Figura 6.34. Se indica además en cada caso el t_{exe} del algoritmo (bastante mayor para el caso del clasificador borroso, como se comenta más adelante en este punto) y el número de objetos $k_{out,t}$ finalmente detectado por el clasificador determinístico a partir de la salida probabilística generada por el XPFPCP.

En la Figura 6.36 se observa un ejemplo de los errores de desplazamiento típicamente producidos por el clasificador borroso (imagen derecha) en situaciones de oclusión parcial. Concretamente, se produce un desplazamiento de 0.5m sobre el eje X en la posición del centroide de la persona a la izquierda de la imagen (cuya clase asociada está marcada en fucsia)

ocultada en gran parte por otra (marcada en verde), y que debido al desplazamiento queda parcialmente fuera de la escena.

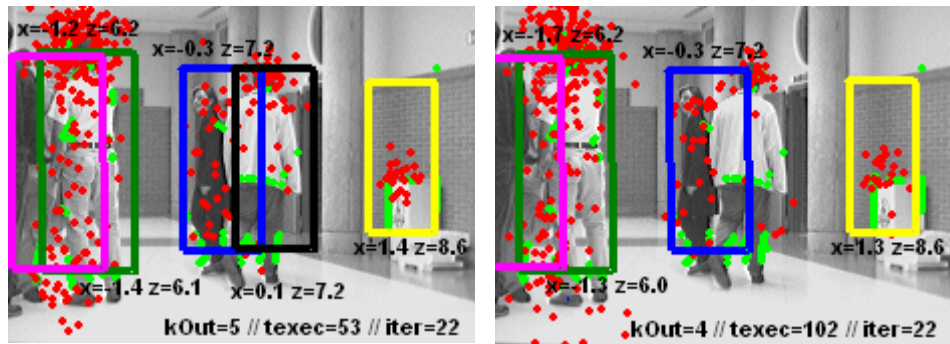


Figura 6.36. Proyección en la imagen de entrada de un experimento real de tracking extraído del test complejo, de los resultados obtenidos al clasificar el set de partículas S_t generado por el XPFCP con un “*k-medias secuencial*” con validación (imagen izquierda), y con un “*subtractive enlazado*” con identificación y validación (imagen derecha). El formato y la información incluida es la de la Figura 6.34, a excepción de la flecha indicativa de la velocidad de cada objeto, eliminada para clarificar la representación.

Sin embargo, el error más evidente que produce el “*subtractive enlazado*” con identificación y validación en el experimento mostrado en la Figura 6.36 es la no generación de la clase (sí generada por el segmentador booleano, como muestra la imagen izquierda, con color negro), asociada a la persona en el centro de la escena (con chaqueta blanca). En este caso, el clasificador borroso ha generado una única clase asociada a las dos personas en el centro de la imagen, debido también a un problema de oclusión.

Conviene finalmente, hacer un comentario sobre los errores generados por ambos segmentadores en la clasificación de partículas, al margen de los de no generación de hipótesis. Si bien la tasa de clases duplicadas generada por el algoritmo booleano es similar a la de clases desplazadas obtenida con el borroso (independientemente de la complejidad de la situación de tracking, como muestra la Figura 6.35), el primer tipo de error tiene una menor repercusión en la aplicación de robótica personal de interés en la tesis (ver en el apéndice III) y, además, puede eliminarse con un post-proceso de la salida generada por el XPFCP.

Para finalizar este apartado, la Figura 6.37 muestra los resultados de clasificación de partículas obtenidos en la ejecución real de los dos algoritmos estudiados (el “*k-medias secuencial*” con validación en la imagen izquierda, y el “*subtractive enlazado*” con identificación y validación en la derecha) en la segunda parte del experimento complejo. En esta figura se representa por colores, en función de su identificador $\tau_{1:4|out}$, la posición el centroide $\bar{g}_{1:4|out}$ de las clases de partículas $G_{1:4|out}$ generadas por los clasificadores. Sobre la secuencia de puntos de color se indica con una flecha el “*ground truth*” manual de posición y dirección del movimiento de los objetos en cada instante del experimento con el que comparan los resultados arrojados por los clasificadores.

Del análisis de la Figura 6.37 se extraen dos conclusiones: el desplazamiento de las clases típico del clasificador borroso se evidencia en la posición de los centroides $\bar{g}_{2|out}$, $\bar{g}_{3|out}$ y

$\bar{g}_{4|out}$; además el clasificador borroso genera un error de identificación (confundiendo la clase $G_{3|out}$ con la $G_{4|out}$ en el momento en que el objeto asociado a la primera oculta al asociado a la segunda) que se mantiene hasta el final del experimento.

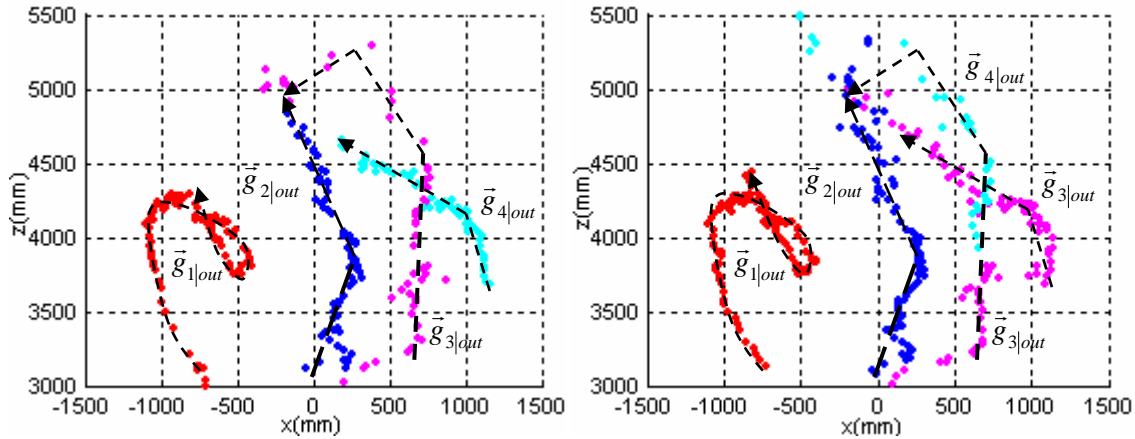


Figura 6.37. Representación por colores de la posición de $\bar{g}_{1:4|out}$, proyectada en el plano XZ , de cada uno de los objetos identificados por el “*k-medias secuencial*” con validación (imagen izquierda) y por el “*subtractive enlazado*” con identificación y validación (imagen derecha) en el set de partículas generado a la salida del XPFCP. Los resultados se han obtenido en la segunda parte del experimento complejo. Al final del experimento el objeto asociado a $G_{3|out}$ oculta totalmente al asociado a $G_{4|out}$, pues pasa antes por el punto de cruce de estas clases.

La confusión de identidad producida por el clasificador borroso se debe (como se comenta en el capítulo 5) a que el proceso de identificación que este incluye no es intrínseco al segmentador, al contrario de lo que sucede con el booleano. Queda así demostrado que la fiabilidad en la identificación de las clases de partículas, es decir, de los objetos implicados en la tarea de seguimiento es, por tanto, mayor al usar el “*k-medias secuencial*” con validación, que al usar el “*subtractive enlazado*” con identificación y validación (un 100% frente a un 89.6% en el ejemplo analizado).

Los análisis de fiabilidad, expuestos en los párrafos anteriores, tienen como conclusión la propuesta del “*k-medias secuencial*” con validación como algoritmo de clasificación de partículas a incorporar a la salida de las cuatro etapas probabilísticas del XPFCP estudiadas en puntos anteriores, y por ende, como generador de la respuesta determinística de estimación multimodal propuesta en esta tesis.

b) Efecto del tipo de algoritmo de clasificación empleado en el tiempo de ejecución del estimador

Se incluye también un análisis de la influencia que la elección de una u otra técnica de clasificación del set de partículas S_t tiene en el t_{exe} global del XPFCP, que es notable, del mismo modo que ocurre con la elección del algoritmo de clasificación de medidas Y_t expuesto en el punto 6.1.3, considerando la importante diferencia que existe en el t_{exe} de las dos propuestas de clasificación presentadas en el capítulo 5 (ver allí punto 5.4.3).

La Figura 6.38 muestra el t_{exe} del XPFCP con clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$, en el experimento complejo, en función del segmentador empleado (gráfica azul para el “*k-medias secuencial*” con validación, y verde para el “*subtractive enlazado*” con identificación y validación). De la figura se extrae la conclusión ya esperada, de que el XPFCP tiene un t_{exe} mayor al usar, para esta tarea, el segmentador borroso que el booleano.

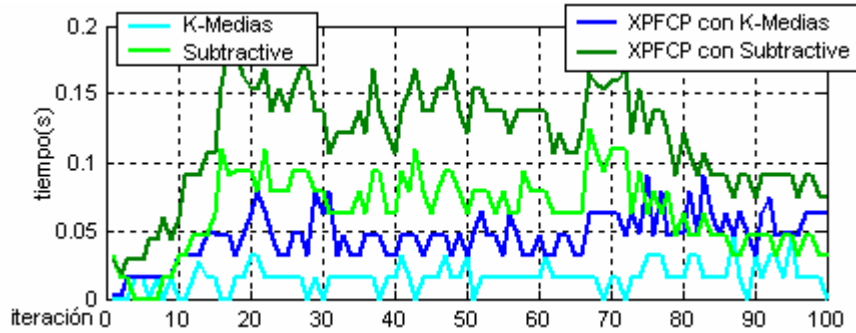


Figura 6.38. Gráficas comparativas del t_{exe} del XPFCP en el experimento complejo, en función del clasificador de partículas usado (con “*k-medias secuencial*” con validación en azul oscuro, y el del clasificador en azul claro, o con “*subtractive enlazado*” con identificación y validación en verde oscuro y el de este segmentador en verde claro).

La Tabla 6.7 (equivalente a la Tabla 6.4) muestra el valor medio del resultado presentado en la Figura 6.38, y permite concluir que, en el test complejo, el t_{exe} del algoritmo global es más del doble al emplear el clasificador borroso que al usar el booleano. En la Tabla 6.7 se incluye también el incremento de t_{exe} que produce la inclusión del algoritmo de clasificación del set S_t en el filtro bayesiano en el mismo test. Este dato permite analizar la carga computacional relativa del clasificador respecto a la global del XPFCP, observándose lo elevado de esta tasa en el caso del segmentador de tipo “*subtractive*”.

Tabla 6.7. Valor medio e incremento absoluto y relativo del t_{exe} del XPFCP al ser ejecutado en el experimento complejo, en función del algoritmo de clasificación de partículas $S_t \Rightarrow G_{1:k,t|out}$ usado.

Clasificador de partículas	t_{exe} total (s)	t_{exe} clasificador (s)	% t_{exe} clasificador del total
“ <i>K-Medias</i> ”	0.045	0.016	35%
“ <i>Subtractive</i> ”	0.104	0.077	74%

Finalmente, la Tabla 6.8 muestra la velocidad media (número de FPS) a la que se ejecuta el seguidor multimodal en las distintas pruebas analizadas (–a– para el test complejo, –b– para el completo corto y –c– para el completo largo), en función del algoritmo de clasificación empleado a la salida del XPFCP, obteniendo en todos los casos un resultado equivalente: la velocidad del XPFCP se duplica si se emplea el segmentador booleano en lugar del borroso para organizar el set S_t que éste genera.

El resultado mostrado en la Tabla 6.8 es casi idéntico en todos los experimentos ejecutados con el clasificador de tipo “*subtractive*”, coincidiendo con el hecho ya comentado de que el t_{exe}

de este segmentador depende fundamentalmente del número de datos a organizar (en este caso, número de partículas a la salida del XPFCP, $n - n_{m,t}$), constante en la propuesta de estimación planteada. La velocidad del XPFCP con un “*k-medias secuencial*” con validación como segmentador de partículas es, sin embargo dependiente del número de clases de partículas $k_{out,t}$, (objetos implicados en la tarea de MTT), y por tanto de la complejidad de la situación de tracking, como confirman los resultados mostrados en la tabla.

Tabla 6.8. Velocidad media de ejecución del XPFCP, en función del algoritmo de clasificación de partículas usado, en distintos experimentos (complejo (a), completo corto (b) y (c) completo largo)

Clasificador de partículas/Test	FPS(a)	FPS(b)	FPS(c)
“ <i>K-Medias</i> ”	22	23.6	25
“ <i>Subtractive</i> ”	9.6	9.6	10

La Tabla 6.8 permite concluir que la propuesta de estimación multimodal cumple la especificación de tiempo real planteada en el capítulo 1, en cualquier caso (la velocidad de ejecución es siempre mayor a 15FPS), si se usa como segmentador de partículas el “*k-medias secuencial*” con validación, propuesto y analizado en el capítulo 5.

c) Efecto del parámetro de distancia

Elegido el “*k-medias secuencial*” como algoritmo para generar la salida determinística del XPFCP, en este punto se analiza el efecto que tiene en la fiabilidad de la salida que éste genera, el valor asignado al parámetro fundamental de su funcionamiento: $distM$.

El análisis aquí incluido es, por tanto, equivalente al mostrado en el apartado 6.1.3, referido en este caso a la aplicación del algoritmo booleano como clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$. Como se recuerda en dicho punto, el funcionamiento del clasificador booleano es bastante sensible al parámetro $distM$, si bien la validación minimiza esta sensibilidad para un rango de valores (alrededor del 30% respecto al punto de ajuste óptimo del parámetro). Por ello es necesario elegir el valor de ajuste de $distM$ de forma teórica.

A partir del planteamiento usado en el punto 6.1.3 para ajustar el mismo parámetro $distM$ del clasificador de medidas $Y_t \Rightarrow G_{1:k,t|in}$, se plantea emplear la matriz de covarianza V , en este caso, del ruido de estado \vec{v}_t , para conocer la dispersión del conjunto de partículas alrededor del centroide de cada clase $G_{j,t|out} / j=1:k_{out,t}$, y determinar así el tamaño que han de tener éstas para incluir todas sus partículas asociadas $L_{j,t|out}$, que define el valor de $distM$.

Para ajustar este parámetro se tiene en cuenta, además, la forma de las clases de partículas $G_{1:k,t|out}$ planteada en el apartado 6.2.1. A partir de este modelo, $distM$ se define como el radio del círculo que incluye el 99.9% de realizaciones del vector de estado en el subespacio XZ , dado por $3 \cdot \sigma_{v,i} / i = \{x, z\}$, aumentado por el radio de dispersión (definido también al 99.9%)

que pueden sufrir las partículas en el subespacio de velocidad $vXvZ$, dado por $3 \cdot t_s \cdot \sigma_{v,i} / i = \{x, z\}$. El valor de ajuste teórico del parámetro queda, por tanto,

$$distM = \sqrt{(3 \cdot \sigma_{v,x})^2 + (3 \cdot \sigma_{v,z})^2} + \sqrt{(3 \cdot t_s \cdot \sigma_{v,x})^2 + (3 \cdot t_s \cdot \sigma_{v,z})^2},$$

expresión que se simplifica teniendo en cuenta el bajo valor de t_s en una aplicación de tiempo real como la de interés:

$$distM = \sqrt{(3 \cdot \sigma_{v,x})^2 + (3 \cdot \sigma_{v,z})^2} = 450mm, \text{ para el valor de } \sigma_{v,i} / i = \{x, z\} \text{ establecido.}$$

El valor fijado de este modo para el parámetro $distM$ se muestra, sin embargo, insuficiente para incluir todas las partículas asociadas a cada clase en los experimentos realizados. Este hecho se debe a que la dispersión de las partículas insertadas en el paso de re-inicialización del XPFCP ($\hat{s}_{i,t-1} / i = n - n_{m,t-1} : n$) se corresponde con la varianza del set de medidas $\sigma_{o,i}^2$, del que éstas se extraen (ver apartado 6.1.1), y no con la del set de partículas $\sigma_{v,i}^2$. Para resolver este problema, el valor de $distM$ se ajusta en función de la mayor de las varianzas, que en este caso coincide con la del conjunto de medidas $\sigma_{o,i}^2$, quedando finalmente:

$$distM_{opt} = \sqrt{(3 \cdot \max\{\sigma_{v,x}, \sigma_{o,x}\})^2 + (3 \cdot \max\{\sigma_{v,z}, \sigma_{o,z}\})^2} \approx 640mm.$$

El otro parámetro de distancia d_{valid} , que define el entorno de error en la predicción del centroide de las clases de partículas $\bar{g}_{1:k,t|out}$ y es necesario en la etapa de validación de la clasificación, se ajusta también siguiendo el planteamiento expuesto en el apartado 6.1.3. Teniendo en cuenta que el modelo de predicción de centroides es equivalente (en la clasificación de partículas se usa el modelo completo CV, en lugar del modificado \widehat{CV}), el valor de ajuste de este parámetro es el mismo para este clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$ que para el de medidas $Y_t \Rightarrow G_{1:k,t|in} : d_{valid,opt} = 425mm$.

d) Efecto del proceso de validación

En este punto se analiza la sensibilidad de los resultados generados por el clasificador de partículas a la salida del XPFCP, en función de los parámetros del algoritmo de validación del “*k-medias secuencial*”, que según los resultados obtenidos en el punto anterior, es el algoritmo elegido para llevar a cabo esta tarea.

Este análisis de sensibilidad del segmentador booleano es similar al incluido en el apartado 6.1.3, pero se repite en este punto debido a las diferentes características del conjunto de datos a clasificar (el set de partículas S_t en este caso, en lugar del de medidas Y_t en la misma aplicación de seguimiento de interés), como ya se comenta en el punto 6.2.1.

La Figura 6.39 muestra la tasa de error generada por el “*k-medias secuencial*” con validación como clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$, en función del valor asignado al parámetro β_{valid} en el experimento complejo. Como muestra la figura, la sensibilidad de la salida determinística que extrae el clasificador no es muy alta (presenta una desviación típica en

la tasa de error de solo el 0.8%, con un cambio del valor de β_{valid} 0 a 5). El análisis de la figura indica, además, que el valor de $\beta_{valid,opt}$ ha de fijarse en el rango de 2 a 4 para asegurar la máxima fiabilidad de la clasificación de S_t , y por ende, del proceso de MTT de interés.

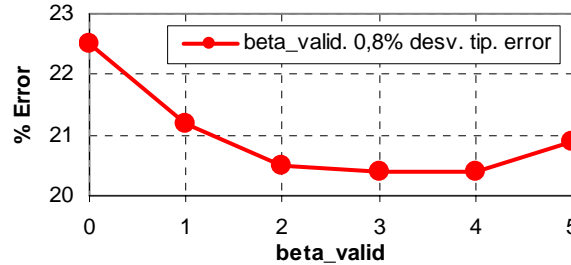


Figura 6.39. Tasa de error obtenida a la salida del clasificador del set de partículas generado por el XPFCP $S_t \Rightarrow G_{1:k,t|out}$, en función del valor del parámetro β_{valid} (beta_valid), en el experimento complejo.

Es importante recordar que la anulación del parámetro β_{valid} conlleva la deshabilitación del proceso de validación, razón por la que aumenta de forma significativa la tasa de error generada por el clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$ en la Figura 6.39. La deshabilitación de este proceso, tiene además el inconveniente de aumentar la tasa de error de identificación de las clases (como se demuestra en el capítulo 5), hecho que en la aplicación del XPFCP aquí analizada significa un aumento del error de identificación de los objetos implicados en el proceso de MTT. Finalmente, el proceso de validación incrementa, también, la tasa de rechazo al ruido del XPFCP, que sin éste coincide con la que genera el filtro sin clasificador de partículas (ver la Tabla 6.9). Por todo ello es importante asignar a β_{valid} un valor no nulo.

Tabla 6.9. Tasa de rechazo al ruido del XPFCP con clasificador de partículas, en el experimento de ruido, en función de la habilitación del proceso de validación incluido.

Valor de β_{valid}	0 (sin validación en cluster de partículas)	2 (con validación en cluster de partículas)
% rechazo al ruido	97.4	99.9

Por otro lado, la Figura 6.40 muestra las gráficas de tasa de error obtenidas del análisis de los resultados generados por el clasificador de partículas a la salida del XPFCP, en el experimento complejo, en función del valor asignado a cada uno de los parámetros de validación normalizados del segmentador: los de validación por verosimilitud en distintos tonos de verde ($p_{valid|max}$ en verde oscuro, α_{valid} en verde intenso y π_p en verde claro) y el de validación por distancia π_d en azul. El valor de ajuste óptimo de cada parámetro analizado en la figura se ha marcado con un círculo del color de la gráfica correspondiente ($p_{valid|max,opt} = \alpha_{valid,opt} = 0.4$, $\pi_{p,opt} = 0.6$ y $\pi_{d,opt} = 0.8$).

En la Figura 6.40 se observa que la sensibilidad del “*k-medias secuencial*” con validación como clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$ al valor de estos parámetros es baja y de magnitud semejante, a la que presenta en su aplicación como clasificador de medidas (ver el

apartado 5.2.6 del capítulo 5). Se confirma así, de nuevo, la robustez de este algoritmo de segmentación en la clasificación del set de salida S_t del XPFCP.

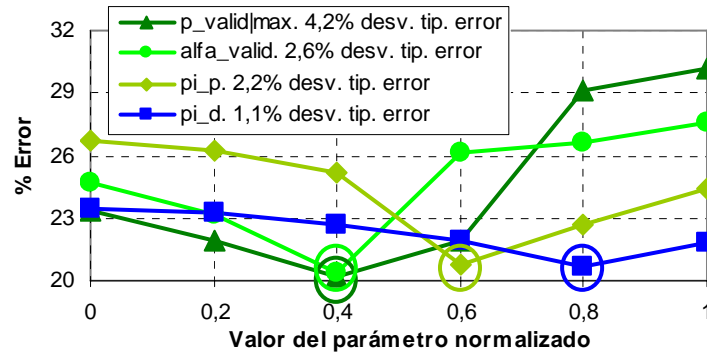


Figura 6.40. Tasa de error obtenida a la salida del clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$, en el test complejo, en función del valor de los parámetros $p_{valid|max}$ ($p_valid|max$), α_{valid} ($alfa_valid$), π_p (pi_p) y π_d (pi_d).

De hecho, la desviación típica del error obtenido al modificar estos parámetros es menor en este caso que la que se obtiene al usar el algoritmo como clasificador de medidas (del 2.5% frente al 3.3% para los cuatro parámetros analizados en la Figura 6.40; y del 0.4% frente al 1.5% para el parámetro β_{valid} ¹³). Este hecho se debe al ya comentado mayor equilibrio en la distribución de las partículas entre las clases $S_t \Rightarrow G_{1:k,t|out}$ y a la propia robustez del XPFCP.

La idea de clasificar el set de partículas a la salida del estimador basado en el PF S_t en aplicaciones de seguimiento aparece en otros trabajos de la literatura del área, como se comenta ya en el capítulo 2. [Vermaak03] es un ejemplo de ello, por lo que en el capítulo siguiente se compara esta y otras propuestas similares con la aquí analizada.

6.2.3. Conclusiones sobre los resultados expuestos

Como comentario final de los análisis expuestos en los puntos anteriores se extraen las siguientes conclusiones sobre la clasificación del conjunto de partículas de salida del XPFCP $S_t \Rightarrow G_{1:k,t|out}$ que, en su aplicación al problema de MTT de interés en la tesis, constituye la solución determinística final:

1. Se demuestra que el “*k-medias secuencial*” con validación, propuesto y descrito en el capítulo 5, tiene un mejor comportamiento como clasificador de partículas que la propuesta borrosa descrita también allí: el XPFCP presenta mayor fiabilidad si se emplea el clasificador booleano en esta etapa de salida (con una tasa de acierto global media del 83.5%) que si se usa el borroso (con el que la tasa de acierto global media baja al 79.3%); genera además un tipo de error menos relevante en la aplicación de interés (el 70% de errores generados por el XPFCP con el clasificador booleano es de no generación o unión de 2 clases en 1 y el 30% restante es por duplicación, mientras que con el borroso el 75% de errores es del primer tipo, y el 25% restante es de desplazamiento, más importantes en la

¹³ Este valor se obtiene de los resultados generados por el clasificador al variar el valor de β_{valid} de 1 a 5.

aplicación de interés, como ya se ha comentado, que los de duplicación); y produce un menor error de identificación de las clases (el XPFCP con clasificador booleano tiene una tasa de acierto del 100% frente a la del 89.6% obtenida si se usa el borroso, en el ejemplo analizado), y por ende, de los objetos implicados en la tarea de seguimiento de interés.

- Además el XPFCP con “*k-medias secuencial*” con validación en su etapa de salida cumple con la especificación de tiempo real establecida, pues tiene un t_{exe} casi 5 veces menor que si se usa el borroso (la ejecución del XPFCP en el primer caso se realiza a una media de 25FPS, en el experimento completo largo, frente a las 10FPS alcanzadas por el mismo en el segundo caso).
- Una vez ajustado el valor de los parámetros del clasificador de partículas $S_t \Rightarrow G_{1,k,t|out}$ ($distM$, β_{valid} , $p_{valid|max}$, α_{valid} , k_{valid} , π_p , d_{valid} , π_d), los resultados obtenidos con él han demostrado ser poco sensibles a la modificación de estos valores, confirmándose así la robustez del segmentador y, por ende, de la solución determinística de seguimiento de múltiples objetos generada por éste a la salida del XPFCP.
- A partir de las características expuestas en el punto anterior, el clasificador booleano propuesto añadido a la salida del XPFCP permite que el proceso de MTT desarrollado con él alcance todas las especificaciones planteadas en el capítulo 1.

Los resultados mostrados en el apartado siguiente corroboran la idoneidad del XPFCP para la tarea de MTT de interés, bajo diferentes condiciones de sensado. Este último apartado del capítulo resume la correcta funcionalidad del seguidor propuesto.

6.3. Resultados globales obtenidos con distintos sistemas de observación

En este último apartado del capítulo se incluye un resumen de los resultados de seguimiento obtenidos con el algoritmo XPFCP propuesto en la tesis a partir de dos sistemas de observación diferentes, basados en cámaras.

El objetivo de este análisis final es el completar la validación del algoritmo de seguimiento de múltiples objetos planteada, mediante una exposición resumida de los resultados globales más importantes alcanzados con él.

Desde este punto de vista, y teniendo en cuenta el marco de aplicación de la propuesta de tracking descrito en el capítulo 1 (la robótica personal), es interesante analizar el comportamiento del XPFCP en un entorno de observación diferente al usado en los análisis empíricos incluidos en este capítulo. Por esta razón se incluye en este punto un primer apartado que resume el comportamiento del XPFCP como algoritmo de tracking a bordo de un robot móvil, y uno segundo en el que se analiza su comportamiento como seguidor en un “*Espacio Inteligente*” ([Lee99], ver en el capítulo 1). En este segundo caso, el procesamiento se distribuye fuera del robot, ubicando los sensores y los sistemas de proceso en su entorno de movimiento (propuesta distribuida), en lugar de embarcados en él (propuesta centralizada).

La problemática del tracking en el segundo de los entornos comentados es algo menor que la de las situaciones analizadas hasta este punto, debido a la existencia de información más completa, e incluso redundante, sobre la posición y, en general, el comportamiento de los objetos que se ubican en cada momento en el entorno, incluido el propio robot ([Pizarro08]). Como contrapartida, el sistema necesario para poner en marcha la aplicación robótica, así como la necesidad de procesamiento para fusionar la información procedente de todos los sensores incluidos en el entorno es mayor.

Como se explica en el capítulo 1, la evolución tecnológica y su presencia cada vez mayor en lugares de desarrollo habitual de las personas, donde la aplicación personal de la robótica tiene su entorno de trabajo (el hogar, oficinas, centros de educación, mercados, museos, estaciones, hospitales, etc.) llevan a aplicar el sistema de MTT propuesto a un “*Espacio Inteligente*”, como comprobación final de la eficacia del XPFCP.

Más adelante, en el punto 6.3.2, se analiza el efecto que tiene el uso de los datos generados por este sistema de observación distribuido como vector de entrada Y_i del XPFCP. Además, en [Pizarro08] se describe de forma detallada la funcionalidad de este sistema así como su aplicación junto al XPFCP al problema de MTT en un “*Espacio Inteligente*”.

6.3.1. Resultados finales obtenidos con un sistema de estéreo-visión ubicado sobre un robot móvil (propuesta de observación centralizada)

En este punto se incluyen los resultados obtenidos en la aplicación del XPFCP a la tarea de MTT a partir de datos de posición obtenidos con el sistema de visión estéreo ubicado sobre un robot móvil (propuesta de observación centralizada), utilizado también en todas las pruebas usadas en este capítulo y el anterior, para validar empíricamente las distintas propuestas planteadas en la tesis. En [Marrón06a] y [Marrón07a] se describe en detalle el funcionamiento, introducido en el capítulo 1, de este algoritmo de detección y extracción de características.

De forma resumida, el funcionamiento de este algoritmo consiste en extraer puntos de borde de los objetos en la escena que pueden ser objeto de seguimiento del algoritmo de tracking. Han de eliminarse, por tanto, los puntos de borde de la estructura que conforma el entorno de movimiento del robot que lleva el sistema de tracking, siendo éste un proceso complejo debido a que el sistema de observación es móvil¹⁴. Una vez conseguidos los puntos de borde, se usa la relación de geometría epipolar entre las cámaras para encontrar la proyección de cada uno de ellos en las dos imágenes y obtener así la posición de estos puntos en el espacio de observación 3D XYZ de interés en la tesis (ver en el capítulo 2).

La Tabla 6.10 resume los resultados globales obtenidos con el XPFCP en con este primer sistema de observación. Los datos son valores medios extraídos de la ejecución del algoritmo en el experimento completo largo, descrito en el apéndice III junto al resto de los referidos en este

¹⁴ Muchos de los trabajos desarrollados por la comunidad científica en el área de tracking en imágenes (como [Pérez02], [Spengler03] o [Branson05]) usan procesos de modelado del fondo de la escena para eliminar de la imagen la información no relacionada con los objetos de interés en la tarea de tracking. Para ello es necesario que la imagen sea estática, cuestión que no se cumple en el sistema de observación aquí descrito.

capítulo. En las primeras entradas de la tabla se resumen las características más importantes de este experimento, así como el valor de ajuste usado para cada parámetro en el test.

Tabla 6.10. Resumen de los resultados globales de seguimiento obtenidos con el XPFCP en la propuesta de observación centralizada (experimento completo largo).

Número de objetos seguidos y tiempo y número total de iteraciones	0-5 objetos, 1098 iteraciones (1min13s)
Valor de ajuste de los parámetros	$n = 600$, $\gamma_t = 0.2$, $n_{init}/n = 5\%$, $distM_{\{in,out\}} = 640mm$, $\sigma_{v,i} = 100/i = \{x, y, z, vx, vz\}$, $\sigma_{o,i} = 150mm/i = \{x, y, z\}$, $\beta_{valid\{in,out\}} = 2$, $d_{valid\{in,out\}} = 425mm$, $P_{valid max\{in,out\}} = 0.4$, $\alpha_{valid\{in,out\}} = 0.4$ $\pi_{p\{in,out\}} = \pi_{d\{in,out\}} = 0.5$ $k_{valid\{in,out\}} = 10$
Fiabilidad (máximo tiempo de fallo y tasa error)	0.6s 3.5%, 0.8s 1.8% El resto de errores son espurios: máximo 200ms (3 iteraciones)
Tasa de error global	12.9%
Tasa de error de no generación de hipótesis	9.2%
Tasa de error de duplicación de hipótesis	3.3%
Tasa de error de desplazamiento de hipótesis	0.4%
Tasa de error de identificación	0%
Tasa de rechazo al ruido	99.9%
Tasa de partículas eficaces	69.8%
Rango de sensibilidad (desv. tip. error al variar parámetros más críticos)	100% varía γ_t o $P_{valid max\{in,out\}}$ genera 3.5%-4% desv. tip. error 50% varía $\sigma_{o,i}$ genera 4% desv. tip. error
Tiempo de ejecución (FPS)	40ms (25FPS)

Los resultados de fiabilidad, sensibilidad, robustez y t_{exe} mostrados en la Tabla 6.10 son coherentes con los que se obtienen con el XPFCP en otros experimentos realizados con el mismo sistema de observación (apartado 6.2).

Merece la pena destacar de la Tabla 6.10 el dato de fiabilidad global obtenida (tercera entrada en la tabla). Este resultado se refiere al número de errores que se producen de forma continua durante un determinado tiempo. Este tipo de errores son los verdaderamente relevantes a la hora de usar la salida del proceso de MTT en una aplicación de robótica personal (en la tarea de navegación, interacción con el usuario, etc.). Como se observa en la tabla, el número de errores que se prolonga por casi 1s no alcanza el 2% (debidas en su mayor parte a oclusiones totales), por lo que se puede decir que la fiabilidad del XPFCP en este test es del 98%.

Es importante también destacar que el XPFCP, debido a su funcionalidad de filtro que permite resolver correctamente muchos de los problemas que aparecen en el sensado (outliers o comportamientos erróneos espurios del sistema de detección), retarda sin embargo la respuesta de tracking al crear nuevas hipótesis de seguimiento. Gran parte de la tasa de error global mostrada en la Tabla 6.10 se debe también a este efecto, como se comenta también en otros

trabajos como [Schulz03b], y se muestra mediante un ejemplo en el apartado siguiente. El valor de fiabilidad mostrado en esta tabla no incluye estos errores, por lo que, desde este punto de vista, este valor también es un indicador más real de la eficiencia del XPFCP propuesto.

Finalmente, la Figura 6.41 muestra una secuencia de pares de imágenes extraídas de la ejecución en tiempo real del XPFCP en el experimento completo largo. Cada par de imágenes (organizado en columna) muestra los resultados obtenidos con el segmentador multi-objeto en dos planos: en el plano imagen en la imagen superior, y en el plano XZ en la imagen inferior.

En la Figura 6.41 se muestran, a modo de ejemplo, los resultados obtenidos en seis frames en pares de imágenes, organizados secuencialmente de izquierda a derecha y de arriba abajo. El formato y la información incluida es la de la Figura 6.34.

Las imágenes de la Figura 6.41 demuestran el correcto funcionamiento del XPFCP propuesto, en situaciones complejas, conclusión a la que se llega del análisis de la figura:

- La posición y velocidad de todos los objetos detectados por el sistema de observación, caracterizados en las imágenes respectivamente por un cilindro o rectángulo (en el plano imagen y XZ, respectivamente) y una flecha (coloreadas en función de la identidad del objeto) es correctamente seguida en todas las situaciones, independientemente de la tipología (forma y dinámica) del objeto. Así, el XPFCP trata por igual en el proceso de tracking tanto a las personas como la papelería detectadas por el sistema de visión a bordo del robot.
- En el frame 258, la Figura 6.41 no muestra el tracking de la papelería, pues ésta no es detectada por el sistema de estéreo-visión usado durante un largo periodo de tiempo. Este hecho sucede varias veces con este sistema de observación, y el XPFCP lo resuelve siempre que el problema no dure más de un número corto de frames, gracias al proceso de validación del clasificador de partículas incluido a su salida, y por supuesto, al efecto de filtrado propio del algoritmo de seguimiento.
- Cada objeto mantiene su identidad mientras permanece en la escena seguida. Por ejemplo, la persona identificada en amarillo cambia su identidad al color azul claro en los dos últimos frames mostrados (354 y 437) al ser eliminado de la escena por estar totalmente oculto durante un largo periodo de tiempo (del frame 427 al 451) y luego volver a aparecer. El XPFCP asigna, por tanto, una nueva identidad a cada objeto que aparece por primera vez en la escena de tracking, como a la persona comentada en este párrafo o a la que aparece instantáneamente a la izquierda de la escena en el frame 190.
- El XPFCP mantiene el seguimiento de los objetos en situaciones de oclusión parcial (presentes en todos los frames excepto el último) y, de forma temporal, en situaciones de oclusión total, como en el frame 437, en el que debido a la geometría epipolar del sistema de estéreo-visión no se obtienen datos de ninguno de los dos objetos seguidos en la escena hasta ese momento (la persona identificada en azul claro, y la papelería), por la oclusión total que genera la persona que pasa justo por delante de las cámaras.

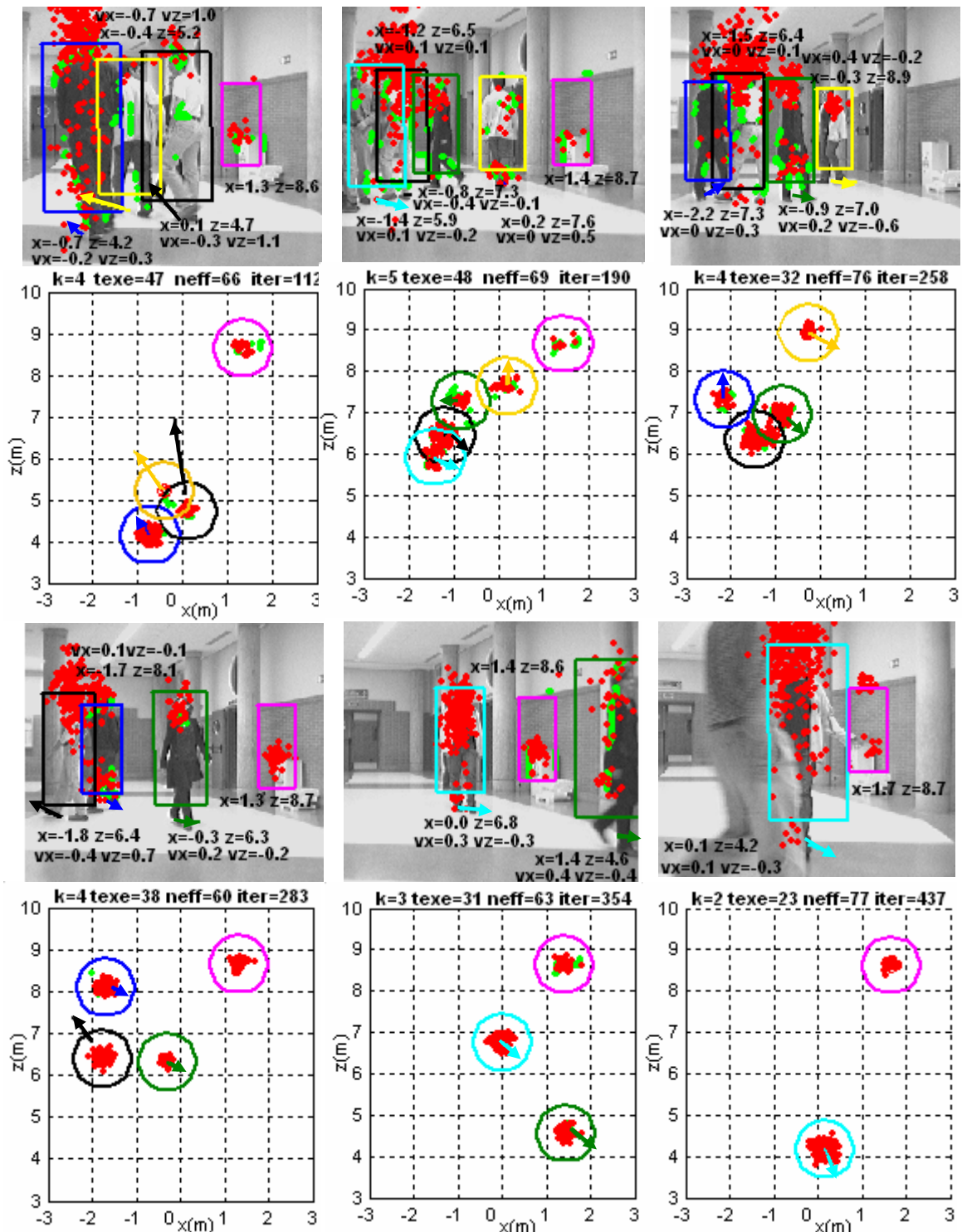


Figura 6.41. Secuencia de pares de imágenes (organizadas en columna y ordenadas de izquierda a derecha y de arriba a abajo) extraídas de la ejecución en tiempo real del XPFCP en la propuesta de observación centralizada (experimento completo largo). Cada par de imágenes incluye la proyección de los resultados obtenidos en dos planos: en el plano imagen en la imagen superior, y en el plano XZ en la imagen inferior. El formato y la información incluida es la de la Figura 6.34.

Los resultados incluidos en la Figura 6.41 muestran, además unos parámetros de t_{exe} y \hat{n}_{eff} adecuados a las especificaciones de diseño planteadas al principio de la tesis, y coincidentes con los expuestos en la Tabla 6.10.

6.3.2. Resultados finales obtenidos con un sistema de visión basado en un anillo de cámaras estático ubicado en el entorno de movimiento del robot móvil (propuesta de observación distribuida)

En este apartado se analizan los resultados obtenidos con el XPFCP a partir del sistema de observación distribuido, introducido en el capítulo 1 y descrito en detalle en [Pizarro08].

De forma resumida, el funcionamiento de este sistema de medida consiste en obtener la proyección en el plano XZ de la zona ocupada en la imagen de cada cámara (ver las imágenes izquierda y central de la Figura 6.42) por los objetos a seguir, mediante un proceso de resta de fondo (típicamente usado en trabajos ya mencionados como [Pérez02] o [Spengler03]); y la posterior intersección de estas imágenes de proyección, a través de la relación geométrica existente entre las cámaras. De este modo se obtiene un conjunto de huellas o blobs (en negro en la imagen derecha de la Figura 6.42) que indican la posición en el plano XZ de los objetos en la escena. El set de puntos que conforman los blobs se usa de vector de medida Y_t de entrada al XPFCP para realizar la tarea de seguimiento.

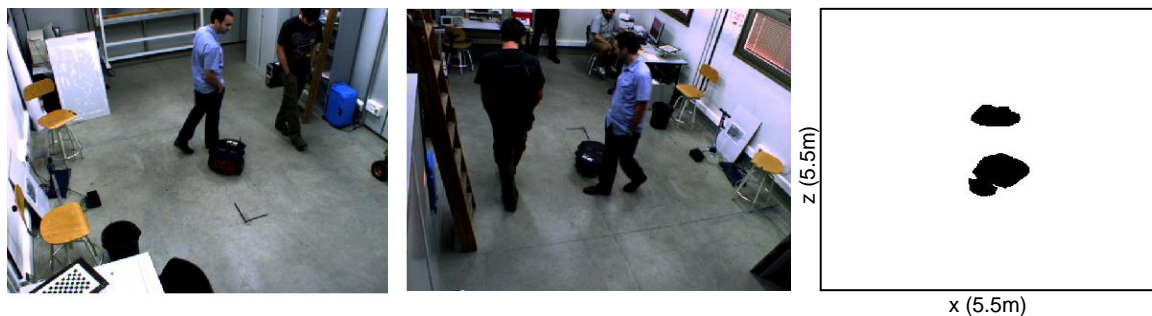


Figura 6.42. Imágenes de una escena real extraídas con el sistema de observación distribuido usado para validar el comportamiento del XPFCP. Las imágenes de la izquierda muestran la escena tal y como se obtiene de dos de las cuatro cámaras de este sistema de observación. La imagen de la derecha muestra el conjunto de blobs obtenidos a la salida de este sistema de observación, y que constituyen el array de medidas de entrada al XPFCP.

El conjunto de medida de posición incluido en Y_t no tiene, por tanto componente de altura, por lo que está definido en el espacio 2D XZ , en lugar de estarlo en el 3D XYZ , planteado como base en los desarrollos expuestos en la tesis. Esta diferencia implica una serie de cambios en la funcionalidad del XPFCP, que se describen a continuación:

1. Ninguno de los algoritmos de clasificación (de medidas y de partículas) incluido en el XPFCP se ve afectado por esta diferencia, pues como se explica en el capítulo 5 y en el apartado 6.2.1, en ningún caso se incluye la altura en el espacio de clasificación.
2. Se modifica el contenido de los vectores de medida y estado, y con ellos el modelo CV usado en el XPFCP (expresión <6.18>), que queda como se muestra a continuación (ecuaciones <6.19> y <6.20>). Esta modificación tiene como única implicación el uso de las nuevas ecuaciones de estado (ecuación <6.19>) y medida (ecuación <6.20>) en los pasos de

predicción y corrección del XPFCP, respectivamente, no afectando a los resultados de tracking obtenidos con el filtro, como se muestra más adelante en este punto.

$$\bar{x}_{t+1} = \begin{bmatrix} x_{t+1} \\ z_{t+1} \\ vx_{t+1} \\ vz_{t+1} \end{bmatrix} = A \cdot \bar{x}_t + \bar{v}_t = \begin{bmatrix} 1 & 0 & t_s & 0 \\ 0 & 1 & 0 & t_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ z_t \\ vx_t \\ vz_t \end{bmatrix} + \begin{bmatrix} v_{x,t} \\ v_{z,t} \\ v_{vx,t} \\ v_{vz,t} \end{bmatrix}, \quad <6.19>$$

$$\bar{y}_t = \begin{bmatrix} x_t \\ z_t \end{bmatrix} = C \cdot \bar{x}_t + \bar{o}_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ z_t \\ vx_t \\ vz_t \end{bmatrix} + \begin{bmatrix} o_{x,t} \\ o_{z,t} \end{bmatrix} \quad <6.20>$$

El XPFCP puede, por tanto, emplearse prácticamente sin modificaciones, en la tarea MTT desarrollada a partir del sistema de observación distribuido.

La Tabla 6.11 resume los datos más importantes del funcionamiento del XPFCP en este caso, obtenidos de los resultados generados por el filtro en un experimento que se describe en el apéndice III (test con observación distribuida), como el resto de los referidos en el capítulo.

Tabla 6.11. Resumen de los resultados globales de seguimiento obtenidos con el XPFCP en la propuesta de observación distribuida.

Número de objetos seguidos y tiempo y número total de iteraciones	2-4 objetos, 1180 iteraciones (1min19s)
Valor de ajuste de los parámetros (se indican los cambios respecto a la Tabla 6.10)	$n = 700$, $\sigma_{v,i} = 120 / i = \{x, y, vx, vz\}$, $d_{valid\{in,out\}} = 510mm$, $p_{valid\{max\{in,out\}} = 0.6$
Fiabilidad (máximo tiempo de fallo y tasa error)	0.6s 2.7% y 1 duplicado de 1.3s y otro de 3.9s El resto de errores son espurios: máximo 200ms (3 iteraciones)
Tasa de error global	10.8%
Tasa de error de no generación de hipótesis	3.8%
Tasa de error de duplicación de hipótesis	6.6%
Tasa de error de desplazamiento de hipótesis	0.4%
Tasa de error de identificación	0%
Tasa de rechazo al ruido	100%
Tasa de partículas eficaces	70.2%
Tiempo de ejecución (FPS)	45ms (22FPS)

La Tabla 6.11 es equivalente a la Tabla 6.10 del apartado anterior, y al igual que en esta última, incluye en sus primeras entradas, las características más importantes del test y el valor modificado (respecto al mostrado en la Tabla 6.10) de los parámetros del XPFCP que han tenido que ser ligeramente reajustados en este caso. Concretamente, los parámetros reajustados son: $p_{valid\{max\{in,out\}}$, debido a la mayor sensibilidad del algoritmo al valor de este parámetro

(como muestra la Tabla 6.10); y $\sigma_{v,i} = 120/i = \{x, y, vx, vz\}$, y con él $d_{valid\{in,out\}} = 510mm$ (el valor de este parámetro se fija a partir del anterior según lo expuesto en el apartado 6.1.2) que se aumenta ligeramente para conseguir una mejor distribución de las partículas en los blobs de medida, y minimizar así el problema de duplicado que la forma de algunas de estas huellas produce a la salida del XPFCP, como se muestra más adelante.

En la Tabla 6.11 se observa que se ha aumentado también en un 16% el número de partículas n utilizado en el XPFCP con sistema de observación distribuida, respecto al número usado con el sistema centralizado (ver la Tabla 6.10). El objetivo de cambio es comprobar, de nuevo, que el aumento de este parámetro no conlleva una mejora sensible de la fiabilidad de la estimación generada por el filtro.

La tasa de error global generada por el XPFCP en este caso (Tabla 6.11) es 2 sólo puntos menor a la obtenida con el sistema centralizado (Tabla 6.10). Este hecho se debe a la mayor sencillez que presenta el problema de MTT al ser observado con este sistema distribuido, como se comenta anteriormente y se demuestra más adelante en este punto. De hecho, la tasa de errores de unión de 2 objetos en 1 disminuye notablemente en este experimento, respecto a la obtenida con el sistema de observación centralizado (se reduce en 6 puntos).

Por otro lado es interesante destacar el hecho de que la mayor parte de la tasa de error global (el 61% de ésta) se debe, en este caso, a la duplicación de hipótesis. La mayor tasa de error de duplicación obtenida con el XPFCP al emplear el sistema de observación distribuido, respecto a la obtenida con el sistema centralizado, se debe, fundamentalmente, al solapamiento de las huellas (datos de posición de entrada al filtro) que produce este método de sensado, como se muestra y analiza más adelante.

El problema de la duplicación de hipótesis queda patente también en el dato de fiabilidad mostrado en la Tabla 6.11. El número de errores que produce el XPFCP de forma continua durante más de 0.6s es equivalente con los dos sistemas de observación (alrededor de un 3.3%) y se debe en ambos casos al retardo, ya comentado, que genera típicamente cualquier algoritmo de tracking ([Schulz03b]) en la creación de nuevas hipótesis de seguimiento. A ese dato de fiabilidad se añaden además, en este caso, dos situaciones de duplicación de larga duración que son las que elevan la tasa global de error.

La Figura 6.43 permite analizar los problemas de tracking comentados en los párrafos anteriores. En ella se muestra el número de objetos por frame en el experimento con observación distribuida generado como solución de tracking por el XPFCP (kXPFCP, en azul), el obtenido como “ground truth” de forma manual (kReal, en verde), y la diferencia entre los dos valores (Error=kReal-kXPFCP, en rojo).

Los picos positivos en la gráfica de error de la Figura 6.43 muestran el retardo producido por el filtro en la creación de hipótesis nuevas, y los dos pulsos negativos de mayor duración (marcados en la figura con línea discontinua) muestran las dos situaciones de duplicación indicadas en la Tabla 6.11. El restante pulso negativo de la gráfica roja se debe a una situación espuria de sensado pobre de uno de los objetos seguidos, que el XPFCP, sin embargo, resuelve correctamente manteniendo su tracking, como demuestra la Figura 6.43.

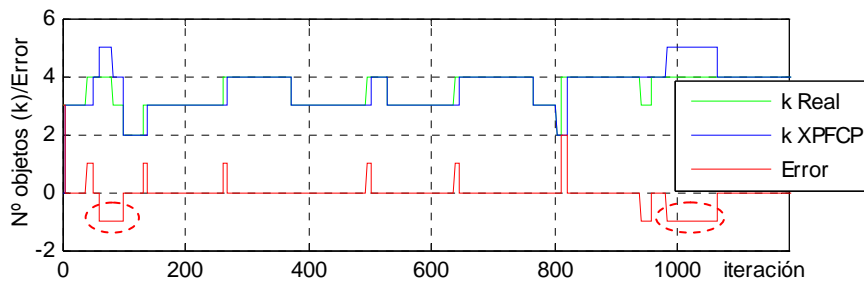


Figura 6.43. Gráfica comparativa del numero de objetos seguido por el XPFCP en la propuesta de observación distribuida (kXPFCP, gráfica azul), frente al observado mediante “ground truth” manual (kReal, en verde) La diferencia entre estos dos valores se muestra también en la figura en rojo (Error=kReal-kXPFCP).

El resto de parámetros de eficiencia del XPFCP mostrados en la Tabla 6.11 (tasa de rechazo al ruido y error de identificación, \hat{n}_{eff} y t_{exe}) son equivalentes a los de la Tabla 6.10, y a los mostrados a lo largo del capítulo, y muestran que el XPFCP cumple las especificaciones planteadas en el capítulo 1, también empleando el sistema de observación distribuido referido.

No se incluye en esta Tabla 6.11 los de sensibilidad del XPFCP al valor de sus parámetros, pues no se ha realizado un estudio tan exhaustivo de este aspecto funcional del filtro con el sistema de observación distribuido, como el expuesto en el apartado 6.1.3 (realizado para el sistema de centralizado). Se ha comprobado, sin embargo, que el XPFCP es sobre todo sensible al parámetro $\sigma_{o,i} / i = \{x, z\}$, cuyo valor que se ajusta, tras analizar la distribución del set de medidas generado por este sistema distribuido, al mismo valor que tiene en la ejecución del XPFCP a partir del sistema centralizado $\sigma_{o,i} = 150mm$.

Finalmente, en este apartado se incluye también una última figura con una secuencia de imágenes extraídas de la ejecución en tiempo real del XPFCP en el entorno de observación distribuido. La Figura 6.44 muestra cinco situaciones de esta ejecución, incluyendo tres imágenes por cada una de ellas organizadas en filas de arriba a abajo: por cada fila, las dos imágenes de la izquierda muestran el resultado del tracking proyectado en el plano imagen de dos de las cuatro cámaras usadas para el sensado distribuido, y la de la derecha la proyección de este resultado en el plano XZ.

La información incluida en la imagen coincide con la mostrada en la Figura 6.41 a excepción del valor estimado de velocidad de cada objeto, eliminado para clarificar la Figura 6.44. A cambio, se dibuja (en el mismo color que identifica a cada objeto) la trayectoria en el plano del suelo seguida por cada objeto, según la estimación generada por el XPFCP.

Como se comenta al principio de este apartado, cada objeto queda representado en las imágenes por un círculo en el plano del suelo en lugar del cilindro usado en la Figura 6.34 (o su proyección en la Figura 6.41), debido al proceso de observación empleado.

Las imágenes de la Figura 6.44, al igual que las de la Figura 6.41, demuestran el correcto funcionamiento del XPFCP, en este caso, en el espacio de observación distribuido. De esta Figura 6.44 se pueden además extraer una serie de conclusiones añadidas a las obtenidas de la Figura 6.41 en el apartado anterior:

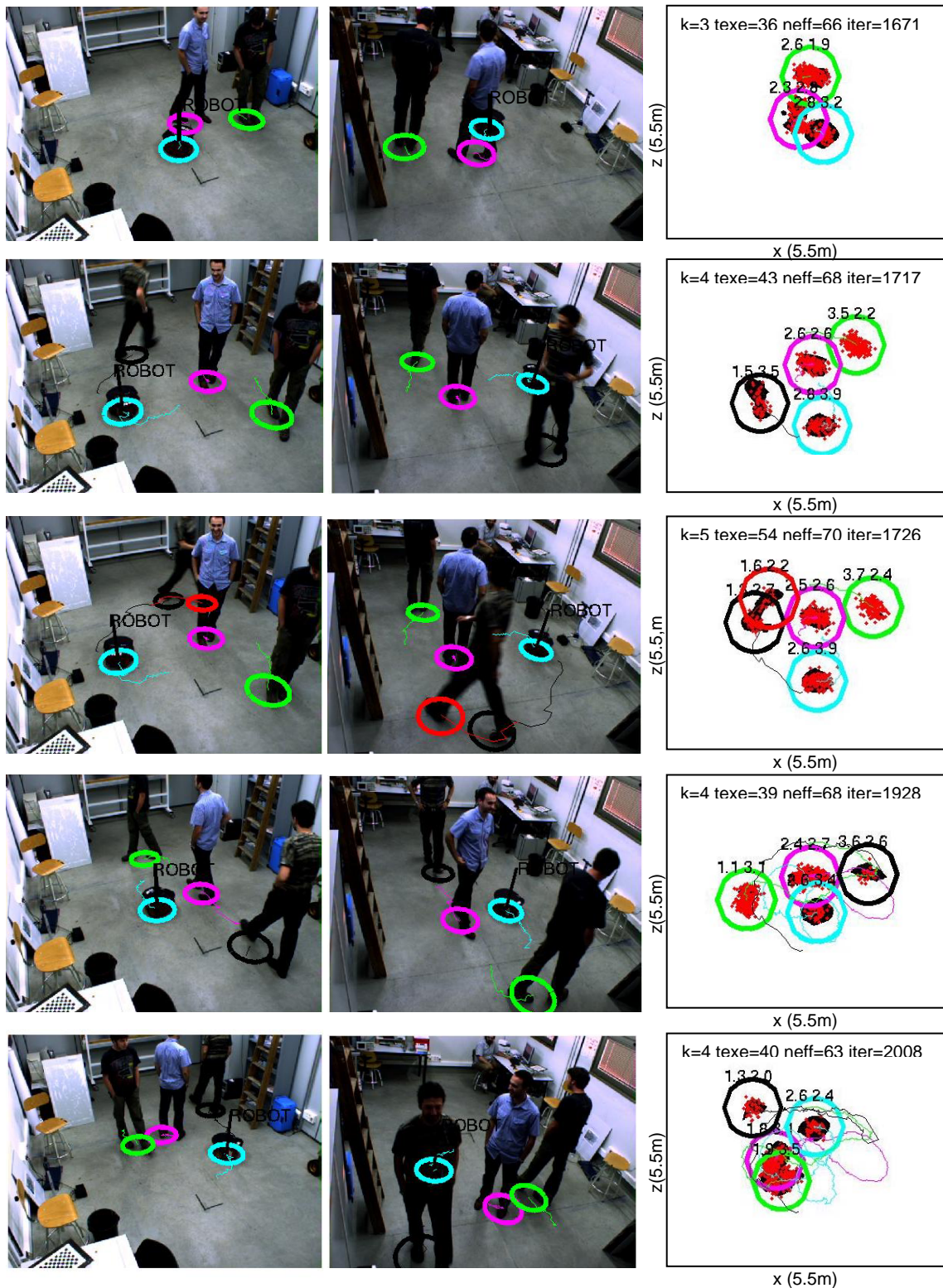


Figura 6.44. Secuencia de tríos de imágenes (organizadas en fila y ordenadas de arriba a abajo) extraídas de la ejecución en tiempo real del XPFCP en la propuesta de observación distribuida. Cada fila de imágenes incluye la proyección de los resultados obtenidos en tres planos (de izquierda a derecha): en el plano imagen de dos de las cuatro imágenes de entrada al filtro, y en el plano XZ . El formato y la información incluida es la de la Figura 6.41, excluyendo la información de velocidad estimada para cada objeto por claridad de la imagen.

- Las imágenes demuestran que el proceso de tracking se simplifica con este método de observación al desaparecer la mayor parte de situaciones de oclusión, causa fundamental de los errores de no generación de hipótesis obtenidos con el sistema de observación centralizado. Las oclusiones parciales (imagen central de la fila primera a la tercera) y totales (imagen central de la última fila) que aparecen en algunas de las imágenes capturadas por las cámaras se eliminan al combinar estas imágenes con las obtenidas del resto de cámaras. De este modo, casi siempre permanece una huella o set de medidas asociado a cada objeto en el array de entrada Y_t al XPFCP, a pesar de las oclusiones.
- Los resultados obtenidos en este experimento vuelven a demostrar (como los de la Figura 6.41) la flexibilidad del sistema de tracking propuesto ante la tipología del objeto a seguir. En este caso se incluye un robot (marcado en azul y con la etiqueta ROBOT) junto a tres personas en la escena en la que se hace el seguimiento. El XPFCP realiza con idéntica fiabilidad el seguimiento de los dos tipos de objetos.
- El sistema de observación distribuido produce, sin embargo errores de tracking, cuando la huella del objeto a seguir se deforma dando lugar a varios blobs, o se solapa con la de otro objeto. En esas situaciones (como la de la huella relacionada con la persona identificada en negro en el frame 1717, que se divide en dos blobs en el frame 1726) el XPFCP mantiene el correcto seguimiento del objeto, pero el clasificador de partículas a su salida duplica la clases asociada a ese objeto (creando una nueva hipótesis, identificada en rojo, también asociada a ese objeto), provocando la relativamente elevada tasa de este tipo de error en la Tabla 6.11. El error de duplicación es, sin embargo, menos importante que otros en la aplicación de interés, como se ha comentado, y puede resolverse con un proceso posterior.

Los resultados incluidos en la Figura 6.44 demuestran, por tanto, la eficacia del XPFCP como algoritmo de MTT en entornos de observación distribuida como el expuesto, y confirman la flexibilidad de la propuesta para funcionar a partir de tipos de medidas de posición diferentes.

6.4. Conclusiones y resumen de las aportaciones realizadas en esta área

Tal y como se comenta en el capítulo 3, y se verifica en éste, en esta tesis se propone el uso de un algoritmo de clasificación determinístico como solución óptima al robustecimiento del estimador de la posición y velocidad de un número variable de objetos basado en un PF.

El PF se modifica, incluyendo en su ejecución iterativa información del conjunto de medidas de entrada al filtro, organizado previamente en clases medidas $Y_t \Rightarrow G_{1:k,t|in}$, de modo que se robustece su funcionamiento multimodal. Las aportaciones incluidas en el flujograma del PF básico que permiten alcanzar esta mejora se resumen a continuación:

1. La re-inicialización del conjunto de partículas a partir del sistema de clasificación de medidas propuesto $Y_{t-1} \Rightarrow G_{1:k,t-1|in}$ hace que la PDF discreta $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$ que genera así el XPFCP se adapte automáticamente para representar el estado $\bar{x}_t^{(j)} / j = 1:k_{out,t}$ del número

variable de obstáculos $k_{out,t}$ presentes en el entorno en cada instante de tiempo, en la aplicación del filtro bayesiano a la tarea de MTT de interés. En este capítulo se demuestra que la multimodalidad del PF no es suficiente para modelar de forma robusta el estado de este número variable de sistemas sin una modificación del algoritmo como la aquí expuesta.

La distribución de las $n_{m,t-1}$ partículas insertadas en el paso de re-inicialización de forma controlada entre las $k_{in,t-1}$ clases de medidas detectadas por el clasificador de entrada favorece al robustecimiento de la creencia discreta $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$, pues permite independizar el funcionamiento del XPFCP de la pobreza del conjunto de medidas a su entrada y del ruido que afecta al sistema de observación del estimador. En este capítulo se demuestra la mejora que el proceso de re-inicialización determinístico propuesto para el XPFCP aporta a la propuesta aleatoria planteada para este paso en [Koller-Meier01].

2. Por otro lado, el uso de la información determinística de clasificación $Y_t \Rightarrow G_{1:k,t|in}$ también en el paso de corrección del XPFCP refuerza el peso $\tilde{w}_t^{(i)} / i = n - n_{m,t-1} : n$ de las partículas $\tilde{s}_{t-1}^{(i)}$ insertadas en el paso de re-inicialización, que de otro modo son eliminadas en el de selección del filtro. En este capítulo se muestra que el algoritmo XPF propuesto en [Koller-Meier01] adolece de este problema, que lo inhabilita para llevar a cabo la estimación multimodal perseguida, y que el problema es resuelto en el XPFCP gracias al proceso de resampling individualizado para cada partícula $\tilde{s}_t^{(i)} / i = 1 : n$ que se propone en esta tesis a partir de la clasificación determinística comentada.

El proceso de estimación bayesiano así generado (con las propuestas de re-inicialización y corrección descritas) permite distribuir de forma equilibrada las partículas del set de salida

del XPFCP $S_t = \left\{ \tilde{x}_t^{(i)}, \frac{1}{(n - n_{m,t})} \right\}_{i=1}^{n - n_{m,t}}$ entre las múltiples hipótesis de seguimiento

representadas por el set en la aplicación del algoritmo a la tarea de MTT de interés, obteniendo una fiabilidad equivalente de todas las hipótesis y robusteciendo el funcionamiento iterativo del estimador multimodal.

3. Finalmente, el set de partículas de salida del XPFCP S_t que representa la solución probabilística de estimación $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$ que éste genera, es también clasificado con objeto de obtener una solución determinística para el problema de MTT al que se aplica el filtro bayesiano propuesto en la tesis. En este capítulo se analiza la robustez, fiabilidad, y coste computacional del algoritmo propuesto para este proceso final, confirmando que el XPFCP resultante alcanza las especificaciones planteadas en el capítulo 1.

La propuesta de estimación multimodal de la tesis para la aplicación de MTT de interés aporta, además, frente a otras, una serie de ventajas indirectas que se enumeran a continuación:

1. Para el proceso de estimación a desarrollar, se plantea el uso del modelo CV muy extendido en el área de aplicación de interés ([Bar-Shalom93], [Vermaak03], [Schulz03b], [Ng06]) debido a su sencillez y baja carga computacional. Además, el empleo del mismo modelo de estimación para todos los targets implicados en la tarea de MTT de interés facilita la tarea

de asociación implícita en esta aplicación. Estas características del modelo elegido favorecen la consecución de la especificación de tiempo real deseada para la aplicación de seguimiento, pero disminuyen la robustez del estimador multimodal, que es reforzada gracias al clasificador de medidas, y del modo expuesto en los párrafos anteriores.

2. El uso del PF en esta tesis como base del proceso de estimación de interés, no se debe a la necesidad de resolver un problema no-lineal, sino debido al carácter multimodal de la estimación que genera. Esta característica del PF permite el uso de un solo estimador para realizar el seguimiento de un número variable de objetos, manteniéndose prácticamente constante, independientemente de la situación, el t_{exe} del proceso de MTT al que se aplica. Este hecho constituye una de las aportaciones principales de la tesis, frente a propuestas basadas en un número variable de estimadores (p. ej. las expuestas en [Schulz03b] o [Vermaak05a]), o en uno solo de vector de estado aumentado (como las mostradas en [Spengler03], [Smith05] o [Ng06]) que tienen un t_{exe} variable, y por tanto, no cumplen la especificación de tiempo real en situaciones complejas de tracking.
3. En este capítulo se demuestra (mediante el análisis del XPF de [Koller-Meier01]) que la versión básica del PF, no es suficientemente robusta para resolver de forma fiable el objetivo de estimación multimodal, debido a lo cual la propuesta de usar un solo estimador multimodal para resolver el problema de interés es abandonada por la comunidad científica
4. El XPFCP aquí expuesto aporta la flexibilidad que le permite ser aplicado a problemas de tracking basados en cualquier tipo de sensor y modelo de observación, e incluso en la fusión de información procedente de varios de ellos. Los resultados mostrados en el apartado 6.3 demuestran esta afirmación, y en [Marron04a] y [Marron04b] se confirma la funcionalidad del XPFCP con un sistema de observación basado en sensores ultrasonidos. Este último sistema tiene un modelo muy distinto a los dos empleados en los test expuestos en apartado 6.3 (y también en [Marrón05a] o [Marrón06b], y [Pizarro08]), por lo que el éxito de la aplicación en las tres situaciones de sensado demuestra la consecución de la característica de flexibilidad perseguida para el XPFCP.
5. La propuesta booleana de clasificación de medidas $Y_t \Rightarrow G_{1:k,|in}$ descrita en el capítulo 5, “*k-medias secuencial*” con validación, demuestra ser, a través de los resultados expuestos, una solución fiable, rápida y robusta al problema de asociación implícito en la tarea de MTT de interés. Este hecho se debe al proceso de identificación de objetos que lleva a cabo el segmentador de forma implícita.

Los análisis y conclusiones expuestos permiten asegurar el correcto funcionamiento del XPFCP como algoritmo de estimación multimodal en su aplicación a la tarea de MTT de interés, alcanzándose además las especificaciones planteadas al principio de la tesis.

En el capítulo siguiente se presenta, como colofón de la tesis, una comparativa del XPFCP aquí descrito frente a los dos algoritmos más extendidos (el JPDAF y el S-JPDAF) en el área de MTT de interés (ver en el capítulo 2), para demostrar el mejor comportamiento de la solución planteada en la tesis desde el punto de vista de fiabilidad, robustez y t_{exe} .

CAPÍTULO 7

COMPARACIÓN TEÓRICO- EMPÍRICA DE LA PROPUESTA

En este capítulo se muestra, como colofón final de la tesis, una comparativa teórica y empírica del XPFCP con otros algoritmos de interés en el área de MTT, con objeto de validar la propuesta de seguimiento realizada en esta tesis.

En primer lugar se incluye el análisis teórico de dos de los trabajos apuntados como más interesantes en el estado del arte de la tesis debido a su semejanza con el XPFCP: el BPF y el MPF. Ambos incluyen uno o más procesos de clasificación para robustecer la funcionalidad del filtro y/o generar una salida determinística a partir de la probabilística obtenida a la salida del PF. La falta de documentación detallada de estos algoritmos en la literatura del área y las diferencias en los problemas de MTT a los que se aplica imposibilita el desarrollo de una comparativa empírica, por lo que, en este caso, el análisis comparativo se limita al plano teórico.

En segundo lugar se analiza el comportamiento de dos algoritmos muy usados por la comunidad científica para resolver problemas de MTT como el aquí planteado: el JPDAF y el S-JPDAF. Se han elegido estos dos algoritmos debido a que ambos se basan en un proceso de asociación de tipo PDA, más extendido que el NN empleado en esta tesis, por su alta fiabilidad ([Rasmussen01]). Además, el segundo de ellos incluye un PF como filtro de estimación, al igual que el XPFCP, mientras que el primero se basa en un KF. De esta forma la comparativa presenta la diversidad necesaria para generalizar las conclusiones que se extraigan, pues se

extiende no solo a propuestas semejantes a la aquí presentada sino a otras con un núcleo de estimación distinto y de probado éxito como el KF.

Tanto el JPDAF como el S-JPDAF han sido implementados y probados en las mismas situaciones de test usadas en el capítulo 6 para analizar el funcionamiento del XPFCP en su aplicación a la tarea de MTT. De este modo se obtiene, al margen de la comparativa funcional, una empírica realizada a partir de los resultados generados por el JPDAF y el S-JPDAF frente a los obtenidos con el XPFCP, en los experimentos descritos en el apéndice III.

Los dos análisis presentados a continuación completan, por tanto, la validación de la propuesta de estimación multimodal para aplicaciones de MTT como la planteada en esta tesis.

7.1. Comparación con propuestas que incluyen clasificación de medidas y/o de partículas

Como se expone en el análisis del estado del arte de esta tesis, la idea de establecer una clasificación $S_t \Rightarrow G_{1:k,t|out}$ dentro del conjunto de partículas que genera el PF aplicado a problemas de MTT, y otra en el de medidas de entrada al mismo $Y_t \Rightarrow G_{1:k,t|in}$, aparece en algunos trabajos realizados por la comunidad científica en este área, de diferentes formas.

- En el caso del clasificador de entrada $Y_t \Rightarrow G_{1:k,t|in}$, en general, el objetivo de la inclusión de un proceso de este tipo en el seguidor multi-objeto en otros trabajos coincide con el perseguido en esta tesis: robustecer el funcionamiento del estimador bayesiano, evitando que el set de partículas degenere, y simplificar el problema de asociación implícito en la tarea de MTT. Sin embargo en ningún caso estas ventajas se usan para aprovechar la multimodalidad del PF para realizar el seguimiento de varios targets con un solo filtro (ver el capítulo 2).
- La incorporación de un clasificador de partículas $S_t \Rightarrow G_{1:k,t|out}$ es menos habitual en las propuestas desarrolladas por la comunidad científica en el marco de la tesis, y en ningún caso el objetivo de su uso coincide con el aquí expuesto: la obtención de una representación determinística del conjunto de objetos seguidos, a partir de la probabilística generada por el filtro no es generalmente necesaria en estos trabajos previos al no estar basados en el modelado de todas las hipótesis de seguimiento con un solo estimador.

De entre las distintas propuestas encontradas en la literatura científica en las dos líneas descritas, destacan las que se analizan a continuación (ya presentadas además en el capítulo 2), por su similitud en varios aspectos con el XPFCP y por ello se eligen para ser comparadas a nivel teórico con el algoritmo de MTT propuesto en esta tesis. La comparativa empírica mostrada en el apartado 7.2 se realiza, como ya se ha comentado, con otros algoritmos más usados por la comunidad científica en la tarea de MTT de interés, que los que se encuadran en este punto.

7.1.1. Clasificación de partículas del “Mixture Particle Filter” (MPF) ([Vermaak03])

En [Vermaak03] se emplea la idea de organizar el set de partículas S_t en $k_{out,t}$ conjuntos, formando lo que los autores denominan “mezcla de componentes” (“mixture components”). Esta clasificación añadida al funcionamiento básico del PF da lugar al “Mixture Particle Filter” o “Filtro de Partículas de Mezcla de Componentes” (MPF). El concepto de “mezcla de componentes” con un objetivo semejante al perseguido en la propuesta de Vermaak se presenta ya en [Gordon97], donde da lugar al “Filtro de Bootstrap Híbrido” (“Hybrid Bootstrap Filter”). En esta publicación preliminar, sin embargo, este concepto no se aplica al robustecimiento de la multimodalidad del PF con objeto de desarrollar la estimación de varios sistemas con un único filtro, por lo que se ha preferido analizar la publicación de Vermaak, en la que el concepto de “mezcla de componentes” se aplica a la tarea de MTT de interés.

En el artículo se demuestra que la suma ponderada (por los denominados “peso de la mezcla j ”, $\pi_{j,t}$) de las componentes es equivalente a la creencia discreta generada por el PF:

$$p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t}) = \sum_{j=1}^{k_{out,t}} \pi_{j,t} \cdot p(\bar{x}_t^{(1:l_{j,t}|out)} | \bar{y}_{1:t}) = \sum_{j=1}^{k_{out,t}} \pi_{j,t} \cdot \sum_{i=1}^{l_{j,t}|out} w_t^{(i)} \cdot \delta(\bar{x}_t^{(i)}), \quad <7.1>$$

donde $l_{j,t}|out$ es el número de partículas asociado con cada $j=1:k_{out,t}$ componente a través de un identificador específico.

En [Vermaak03] se propone usar una distribución gaussiana para cada componente $p(\bar{x}_t^{(1:l_{j,t}|out)} | \bar{y}_{1:t}) = norm(\bar{x}_{j,t}, P_{j,t}) / j=1:k_{out,t}$, de modo que su media $\bar{x}_{j,t}$ sea la solución determinística de la posición de cada target en la aplicación del MPF a la tarea de MTT.

Los resultados mostrados en el artículo se obtienen usando como proceso de reagrupación o clasificación de partículas un “ k -medias”¹, hecho que coincide con la propuesta realizada en esta tesis, al igual que el empleo de un modelo CV en la estimación de posición. La relación entre el XPFCP y el MPF es mayor si se identifica el peso de cada componente del MPF $\pi_{j,t} / j=1:k_{out,t}$ con la probabilidad de cada clase $p_{j,1:t|out}$ a la salida del XPFCP, pues en [Vermaak03], este valor $\pi_{j,t}$ está relacionado con el número normalizado de partículas de cada componente $\tilde{l}_{j,t}$, de forma análoga al valor de probabilidad de las clases $p_{j,1:t|out}$ en el XPFCP.

Sin embargo, el MPF muestra también ciertas diferencias, enumeradas a continuación, frente al XPFCP, que permiten prever una mayor robustez y fiabilidad de la propuesta presentada en esta tesis frente a la expuesta en [Vermaak03]:

¹ Se supone que en [Vermaak03] se propone el uso de alguna versión modificada del “ k -medias”, como el “ k -medias extendido”, para clasificar el set de partículas, pues si bien esto no se especifica en el artículo, sí se expone allí la posibilidad de que el número de componentes (o clases) a la salida de la reagrupación sea diferente al de su entrada.

1. El objetivo de insertar un clasificador que agrupe las partículas en componentes en el MPF, es obtener un conjunto de PFs prácticamente independientes cuyas partículas se propaguen por el paso de predicción, corrección y selección de forma independiente para cada hipótesis de tracking (cada objeto seguido), asegurando así la multimodalidad de la creencia discreta generada a la salida del filtro. El MPF funciona, por tanto, como un conjunto de PFs casi independientes (como el seguidor propuesto en [Schulz03b]), pues sólo interactúan en el proceso de reagrupación, en el que las partículas pueden cambiar su asignación de una a otra componente. El MPF no cumple, por tanto, el objetivo planteado en la tesis de emplear un solo estimador con número de partículas constante para desarrollar el seguimiento de un número variable de targets en tiempo real y constante.
2. La degeneración del set en el MPF se evita, por tanto, al realizar el resampling de las partículas de forma independiente para cada componente. Este procedimiento imposibilita el empobrecimiento natural de las componentes, necesario, por ejemplo, para eliminar un modo o hipótesis de la creencia discreta a la salida del filtro cuando el objeto al que se asocia desaparece de la escena. En el XPFCP la multimodalidad de la creencia de salida del filtro queda asegurada con la implementación de los pasos de re-inicialización y corrección en función del resultado generado por el clasificador de medidas $Y_t \Rightarrow G_{1:k,t|in}$, como se explica en el capítulo 6. Esta solución permite, además, incorporar y eliminar nuevas hipótesis de estimación en esta única creencia discreta de forma automática, cuestión no considerada en [Vermaak03].
3. De la lectura de [Vermaak03] se supone que la eliminación de las partículas de componentes que representan a objetos que desaparecen de la escena (así como la asignación de partículas a componentes nuevas que representen a los objetos que aparecen en la escena) la realiza el propio proceso de reagrupación de componentes. Sin embargo, en ningún punto del artículo se especifica este procedimiento que, en cualquier caso no se realiza de forma automática como en el XPFCP.
4. Por otro lado, la función de pesado de las partículas de cada componente $w(\bar{x}_t^{(i)})$ del MPF, así como el proceso de asociación que normalmente la define no se explica claramente en [Vermaak03], si bien la ponderación se realiza independientemente para cada componente. En el XPFCP la asociación entre cada partícula y la verosimilitud correspondiente se hace parcialmente de forma implícita en el proceso de clasificación de medidas $Y_t \Rightarrow G_{1:k,t|in}$, y sin necesidad de independizar los distintos modos de la creencia en función de las hipótesis. De este modo se robustece la funcionalidad del estimador, a la vez que se facilita el refuerzo o la debilitación automática de las hipótesis representadas por el set S_t a la salida del filtro.
5. El XPFCP no restringe la representación de las hipótesis de seguimiento a su salida a una forma gaussiana, como ocurre con el MPF, por lo que presenta mayor flexibilidad que el anterior en su aplicación al problema de MTT de interés.

Ni en [Vermaak03], ni en trabajos posteriores de sus autores, se presentan resultados empíricos exhaustivos, por lo que se desconoce, entre otros valores, el t_{exe} del MPF aplicado a la tarea de MTT. Los resultados mostrados en el artículo no pueden ser, por tanto comparados

con los obtenidos con el XPFCP en el capítulo 6, y, a pesar de las semejanzas funcionales que presentan, la falta de definición del MPF imposibilita su implementación. Este hecho lleva a usar el S-JPDAF descrito en [Schulz03b] en la comparativa empírica incluida más adelante, basado en un concepto similar al de “*mezcla de componentes*” propuesto en [Vermaak03].

Como se comenta en el capítulo 2, en [Vermaak05b] se aportan varias mejoras al algoritmo de seguimiento multi-objeto. Estas modificaciones no se realizan, sin embargo, sobre el MPF, sino sobre una de tipo JPDAF semejante al descrito en el apartado 7.2.1 (basada en el uso de un KF por target como núcleo de estimación del seguidor multi-objeto). El algoritmo resultante gestiona con éxito el tracking de un número variable de objetos, pero presenta problemas de ejecución en tiempo real (como se demuestra más adelante).

En otros artículos del mismo grupo de trabajo de la Universidad de Cambridge, como [Ng06], se plantea la inclusión de un proceso específico de creación (“*birth*”) y eliminación (“*death*”) de hipótesis de estimación en el marco del PF. A diferencia del método propuesto en [Vermaak03] con este mismo objetivo, este algoritmo se describe en detalle en el artículo y se basa en la clasificación del set de medidas a la entrada del filtro. Antes de ser efectivas en la creencia discreta representada por el set de partículas a la salida del filtro, estas propuestas de modificación han de pasar por un proceso de validación por ventanas temporales semejante al propuesto en [Blackman99] (ver en el capítulo 2), y a la validación incluida en las propuestas de clasificación presentadas en esta tesis (y con anterioridad en [Marrón04a], [Marrón04b] o [Marrón05a]). En [Ng06] se explica que esta mejora se incluye en un algoritmo tipo S-JPDAF de vector de estado extendido en lugar de en el MPF propuesto en [Vermaak03]. Debido a ello, ni en [Ng06] ni en ningún artículo posterior de sus autores se presenta una aplicación en tiempo real del algoritmo así obtenido.

Las propuestas planteadas por este grupo de investigación no responden, por tanto, a las especificaciones planteadas en esta tesis para el desarrollo de la aplicación de MTT de interés, si bien presentan varias ideas comunes con las aquí presentadas.

7.1.2. Clasificación de medidas del “*Boosted Particle Filter*” (BPF) ([Okuma04])

El trabajo presentado en [Okuma04] tiene al igual que el analizado en el punto anterior una especial relevancia en esta tesis, como se comenta en el capítulo 2. En él, los autores plantean la incorporación de un detector “*Adaboost*” ([Viola01]) al MPF propuesto en [Vermaak03] y explicado en el punto anterior, dando lugar al “*Boosted Particle Filter*” (BPF).

El detector “*Adaboost*” clasifica la información de entrada al filtro Y_t , del mismo modo que el segmentador de medidas propuesto en esta tesis. El algoritmo “*Adaboost*” está formado por un conjunto de clasificadores en cascada que, en aplicaciones de visión, permiten detectar si la imagen contiene ciertas características de un patrón de entrenamiento. En el proceso descrito en [Okuma04] este algoritmo se entrena para detectar jugadores de jockey en una pista de hielo (ver en el capítulo 2) a partir del histograma de color de la imagen. La salida del detector se usa para obtener una función de verosimilitud específica para cada clase de medidas

$$p(\bar{y}_t^{(j)} | \bar{x}_t) / j = 1 : k_{in,t}.$$

Esta información obtenida con el detector “*Adaboost*” se incorpora al MPF descrito en [Vermaak03], eliminando algunas de las carencias (analizadas en el punto anterior) que presenta dicho estimador bayesiano, según se explica en [Okuma04] y se resume a continuación:

1. Las funciones de verosimilitud específicas $p(\bar{y}_t^{(j)}|\bar{x}_t)/j=1:k_{in,t}$ son usadas en la ponderación de las partículas de cada componente del BPF. Además, para completar la definición del proceso de pesado, se define una función de aproximación a la creencia también específica para cada conjunto de partículas que conforman la componente, que depende del proceso de asociación entre ésta y el conjunto Y_t , de la siguiente forma:

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot \frac{p(\bar{y}_t^{(j)}|\bar{x}_t) \cdot p(\bar{x}_t|\bar{x}_{t-1})}{q_{BPF}(\bar{x}_t|\bar{x}_{0:t-1}, \bar{y}_{1:t})} = w_{t-1}^{(i)} \cdot \frac{p(\bar{y}_t^{(j)}|\bar{x}_t) \cdot p(\bar{x}_t^{(i)}|\bar{x}_{t-1})}{\alpha \cdot q_{ada}(\bar{x}_t^{(i)}|\bar{x}_{0:t-1}, \bar{y}_{1:t}) + (1-\alpha) \cdot p(\bar{x}_t^{(i)}|\bar{x}_{t-1})}, \langle 7.2 \rangle,$$

donde $i=1:n$, $j=1:k_{in,t}$ y α es un parámetro que permite modular linealmente el tipo de función de aproximación $q(\bar{x}_t|\bar{x}_{0:t-1}, \bar{y}_{1:t})$ usada en la ponderación de cada partícula, haciendo: que ésta sea la PDF de actuación $p(\bar{x}_t^{(i)}|\bar{x}_{t-1})$ de la partícula, en caso de que el detector “*Adaboost*” no genere una observación en $G_{1:k,t|in}$ relacionada (a través de una asociación de tipo NN) con la componente a la que la partícula está asignada ($\alpha=0$); que sea la PDF $q_{ada}(\bar{x}_t^{(i)}|\bar{x}_{0:t-1}, \bar{y}_{1:t})$ que relaciona la partícula con la clase de medidas $G_{j,t|in}$ asociada, en caso contrario ($\alpha=1$); o que sea una combinación las 2 densidades ($0 < \alpha < 1$).

El procedimiento de ponderación es, por tanto, equivalente al propuesto en esta tesis, pues como muestra la expresión $\langle 6.11 \rangle$ del capítulo 6, el peso $w_t^{(i)}$ de cada partícula $i=1:n$ en el XPFCP se obtiene a partir de la PDF gaussiana $p(\bar{g}_{1:k,t|in}|\bar{x}_t^{(i)})$ específica para cada una de ellas, y que depende de la relación (dada también por la asociación NN) entre la hipótesis de estimación representada por la partícula y la clase de medidas $G_{j,t|in}/j=1:k_{in,t}$ más semejante a ésta en el espacio de observación.

2. La incorporación de la información generada por el algoritmo “*Adaboost*” al paso de corrección del MPF mejora la robustez del pesado de las partículas, y por ende (como demuestran los resultados mostrados en el capítulo 6), disminuye el problema de empobrecimiento típico de las versiones básicas del PF, y permite, por ende, que la creencia generada a la salida del filtro $p(\bar{x}_t^{(1:n)}|\bar{y}_{1:t})$ mantenga su multimodalidad sin recurrir al resampling independiente del set de partículas asociado a cada componente, propio del MPF. Sin embargo, el BPF mantiene el desarrollo intra-componente del paso de selección del MPF, por lo que funciona, al igual que este último, como un conjunto de estimadores casi independientes. El BPF no cumple, por tanto, el objetivo planteado en la tesis de llevar a cabo la estimación del estado de un número variable de sistemas con un solo filtro.
3. Por otro lado, el detector “*Adaboost*” se emplea para crear componentes nuevas en la mezcla de partículas generada a la salida del BPF $p(\bar{x}_t^{(1:n)}|\bar{y}_{1:t})$, en el caso de que se

detecten nuevas hipótesis de seguimiento (nuevos objetos a seguir) en la imagen. En [Okuma04] no se explica cómo se realiza esta modificación de la creencia discreta $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$, y tampoco se especifica la adición de ningún paso específicamente para ello en el lazo de ejecución del BPF, como el de re-inicialización del XPFCP. Se asume, por tanto, que la modificación se consigue generando nuevas partículas que son asociadas a la nueva componente, mediante un algoritmo típico de gating (ver en el capítulo 2). Esta suposición implica que el número de partículas del BPF cambia en función del número de hipótesis de tracking gestionadas por él, en contra de lo especificado en esta tesis a partir del objetivo de ejecución en tiempo real establecido.

Ni en [Okuma04] ni en trabajos posteriores de sus autores, se encuentra una definición específica de la función $q_{ada}(\bar{x}_t^{(1:n)} | \bar{x}_{0:t-1}, \bar{y}_{1:t})$, por lo que, como ocurre con el MPF antes analizado, no es posible implementar el algoritmo allí descrito para compararlo empíricamente con el XPFCP propuesto en esta tesis y se recurre al uso del S-JPDAF ya comentado. En el artículo tampoco se incluye información sobre los resultados obtenidos en la aplicación en tiempo real del BPF al problema de MTT de interés, por lo que el algoritmo tampoco es comparable desde este punto de vista con el propuesto en la tesis.

En [Kobayashi06] se presenta un algoritmo para aplicaciones del MTT como la de interés muy semejante al BPF. La diferencia fundamental de la propuesta presentada en ese artículo frente a la descrita en [Okuma04] se encuentra en el modo de usar la salida del clasificador “*Adaboost*” en el proceso de ponderación de las partículas. En [Kobayashi06], sin embargo, sólo se muestran resultados de seguimiento en tiempo real de un único objeto, que por tanto, tampoco son comparables con los alcanzados con el XPFCP propuesto en esta tesis.

Es importante destacar que propuestas como las descritas en [Okuma04], [Ng06] o [Kobayashi06] confirman la teoría planteada en esta memoria y en trabajos publicados por la autora al tiempo de los anteriores ([Marrón04a], [Marrón04b] o [Marrón05a]) de que la incorporación en el peso de cada partícula, de información de observación filtrada por un clasificador y relacionada con la hipótesis de estimación a la que ésta representa, mejora la eficiencia, robustez y fiabilidad de la estimación basada en un PF. Se evitan, de este modo, los problemas de empobrecimiento característicos de este filtro y se refuerza la perseguida multimodalidad de la creencia discreta $p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$ representada por el set S_t .

Los algoritmos usados en la comparativa empírica con el XPFCP, que se analiza en el apartado siguiente, no incorporan, sin embargo ningún proceso de clasificación de medidas, por lo que no incluyen las ventajas comentadas en el párrafo anterior.

7.2. Comparación con otras propuestas típicas de la aplicación de seguimiento de múltiples objetos

En este apartado se concluye el análisis comparativo del XPFCP con otras propuestas de interés en el área de seguimiento de múltiples objetos.

Se incluye, en primer lugar, la descripción funcional de los dos algoritmos elegidos para realizar la comparativa empírica y funcional: el JPDAF ([Bar-Shalom88]) y una versión muestreada de éste (basada en un PF), el S-JPDAF ([Schulz03b]).

En ambos casos, el algoritmo tiene como núcleo un estimador probabilístico organizado en los dos pasos básicos de la regla de Bayes (predicción y corrección), y una etapa de asociación, incluida en el segundo de estos pasos, que permite definir la relación entre el conjunto de medidas de entrada al estimador Y_t y alguna de las $k_{out,t}$ hipótesis implicadas en la estimación.

El algoritmo de asociación incluido tanto al JPDAF como al S-JPDAF es un JPDA ([Fortmann83]). En el capítulo 2 se explica que este proceso resuelve de forma probabilística el complejo problema de asociación con múltiples hipótesis que aparece en aplicaciones como la de interés. El JPDA es el algoritmo de asociación tradicionalmente más utilizado en tareas de MTT, como demuestran trabajos como [Rasmussen01], [Schulz03b], [Vermaak05b] o [Ng06], sin embargo su alto coste computacional dificulta su ejecución en tiempo real.

Este hecho se confirma analizando los resultados, incluidos en segundo lugar en este apartado, que se obtienen al ejecutar ambos estimadores (el JPDAF y el S-JPDAF) en los tests de MTT usados para validar la funcionalidad del XPFCP en el capítulo 6.

Los resultados expuestos en los puntos que siguen permiten, además, comparar la fiabilidad, sensibilidad y robustez del JPDAF y del S-JPDAF frente a las del XPFCP propuesto, confirmándose el hecho de que la propuesta de estimación presentada en la tesis, a pesar de incluir un proceso de asociación determinístico teóricamente menos fiable y robusto que el JPDA ([Rasmussen01]) obtiene mejores resultados en la tarea de MTT de interés que los que genera cualquiera de los otros dos algoritmos.

Por otro lado, el JPDAF incluye como núcleo de estimación un KF, mientras que el S-JPDAF incluye un PF. Estos filtros (el KF y el PF) son las dos alternativas de estimación más usadas en el área de MTT debido a su carácter probabilístico, razón por la que la elección del JPDAF y del S-JPDAF para realizar la comparativa mencionada aporta la diversidad necesaria para validar la funcionalidad y eficiencia del XPFCP propuesto.

Finalmente, conviene resaltar que tanto el JPDAF como el S-JPDAF se comportan como estimadores unimodales en la aplicación aquí expuesta, pues se basan en el uso de un filtro por cada target a seguir. De este modo los dos algoritmos presentan una funcionalidad muy diferente a la del XPFCP propuesto, cuya aportación principal es, como ya se ha comentado, emplear una sola PDF multimodal para definir el estado de los múltiples targets seguidos.

En artículos como [Schulz03b], en el que se defiende el uso del S-JPDAF para la aplicación de interés en la tesis, se asegura que la propuesta multimodal planteada en otros trabajos como el XPF de [Koller-Meier01] previos a la tesis, es computacionalmente más adecuada para la aplicación de MTT, pero no tiene suficiente robustez. En esta tesis se proponen técnicas que resuelven este problema, y se demuestra que el algoritmo resultante, el XPFCP, alcanza la especificación de robustez necesaria para su uso en aplicaciones de robótica personal.

7.2.1. El “Filtro de Asociación Probabilística Conjunta de Datos” (“Joint Probabilistic Data Association Filter”, JPDAF)

El JPDAF es un estimador diseñado fundamentalmente para aplicaciones de MTT en el que se combina: la formulación probabilística del JPDA para obtener, con ciertas restricciones, el conjunto de valores de probabilidad de asociación conjunta $p_{i,j,t} / i=1:m_t, j=1:k_{t,out}$ de las distintas hipótesis de seguimiento, a partir de todas las posibles soluciones de asociación Φ ; con la formulación óptima de la regla de Bayes para estimar el estado de estas hipótesis $\bar{x}_{j,t}$, en función de las asociaciones.

Desde la primera de sus implementaciones teóricas ([Bar-Shalom88]), el KF se ha usado como formulación de la regla de Bayes en el JPDAF, por lo que para obtener un funcionamiento óptimo en su aplicación al problema de seguimiento es necesario establecer un conjunto de restricciones en la definición de los modelos de observación y actuación de cada uno de los $k_{out,t}$ targets implicados en el problema.

El JPDAF se define, por tanto, como la unión de un KF y un JPDA para resolver de forma probabilística el problema de MTT, bajo las condiciones que se listan a continuación.

- Los modelos de observación y actuación de los $k_{out,t}$ targets implicados han de ser lineales y estar afectados por ruidos gaussianos, blancos, de media nula, e incorrelados, de modo que se pueda aplicar la definición básica del KF, descrita en el capítulo 2 ([Maybeck79]).
- El número de targets o hipótesis de estimación ha de ser constante e incluir una que caracterice al clutter ([Bar-Shalom88]). Esta hipótesis se emplea para gestionar los outliers en el proceso de asociación JPDA. Además, en cada solución de asociación Φ solo puede haber una medida asociada con cada target, y cada medida solo puede además asociarse con un único target.

Con todo ello, el JPDAF permite obtener los valores de probabilidad $p_{i,j,t} / i=1:m_t, j=1:k_{t,out}$, a emplear en el paso de corrección de cada uno de los $k_{out,t}$ KFs usados para obtener la estimación del estado $\bar{x}_{j,t}$ de cada hipótesis de seguimiento (se emplea un KF por cada hipótesis de seguimiento incluida en la aplicación de MTT).

Para aplicar el JPDAF a la tarea de tracking de interés en esta tesis, se propone usar el modelo CV modificado (\widehat{CV}) propuesto en el capítulo 5 (ecuaciones <5.5> y <5.6>) para el proceso de predicción del valor inicial de los centroides. De este modo se simplifica el cálculo matricial implicado en el JPDAF², alcanzándose idéntica fiabilidad en la estimación de la posición de cada objeto o hipótesis de seguimiento, a costa de renunciar a la obtención de un valor estimado de su velocidad, que no es generado por el KF así implementado al haber eliminado estas componentes del vector de estado en el modelo \widehat{CV} (ver la ecuación <7.3>).

² En el modelo $\widehat{CV} C$ se transforma en una matriz identidad, como muestra la expresión <7.5> posterior, lo que simplifica los cálculos implicados en el paso de corrección del KF, mostrados en la ecuación <2.14> del capítulo 2.

Las ecuaciones de actuación (<5.5> del capítulo 5) y observación (<5.6>) de este modelo se repiten a continuación para facilitar la explicación funcional del algoritmo y analizar el cumplimiento de las restricciones impuestas para la aplicación del KF a estos modelos.

$$\bar{x}_{j,t} = \begin{bmatrix} x_{j,t} \\ y_{j,t} \\ z_{j,t} \end{bmatrix} = A \cdot \bar{x}_{j,t-1} + B \cdot \bar{u}_{j,t-1} + \bar{v}_{j,t-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{j,t-1} \\ y_{j,t-1} \\ z_{j,t-1} \end{bmatrix} + \begin{bmatrix} t_s & 0 \\ 0 & 0 \\ 0 & t_s \end{bmatrix} \cdot \begin{bmatrix} vx_{j,t-1} \\ vz_{j,t-1} \end{bmatrix} + \begin{bmatrix} v_{x,j,t-1} \\ v_{y,j,t-1} \\ v_{z,j,t-1} \end{bmatrix}, \quad <7.3>$$

$$\bar{y}_{j,t} = \begin{bmatrix} x_{j,t} \\ y_{j,t} \\ z_{j,t} \end{bmatrix} = C \cdot \bar{x}_{j,t} + \bar{o}_{j,t} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{j,t} \\ y_{j,t} \\ z_{j,t} \end{bmatrix} + \begin{bmatrix} o_{x,j,t} \\ o_{y,j,t} \\ o_{z,j,t} \end{bmatrix} \quad <7.4>$$

El subíndice $j=1:k_{out,t}$ en las expresiones <7.3> y <7.4> identifica al objeto al que se refiere el modelo, siendo este último idéntico para todos los objetos implicados en la estimación al emplear el mismo planteamiento usado en la aplicación del XPFCP a este problema.

Las ecuaciones <7.3> y <7.4> se simplifican obteniéndose la forma definitiva del modelo \widehat{CV} para cada objeto $j=1:k_{out,t}$ (expresión <5.8> del capítulo 5):

$$\begin{aligned} \bar{x}_{j,t} &= \bar{x}_{j,t-1} + B \cdot \bar{u}_{j,t-1} + \bar{v}_{j,t-1} \\ \bar{y}_{j,t} &= \bar{x}_{j,t} + \bar{o}_{j,t} \end{aligned} \quad <7.5>$$

En la ecuación <7.5>, las señales $\bar{v}_{j,t}$ y $\bar{o}_{j,t}$ caracterizan, como se comenta en capítulos anteriores, a las variables aleatorias de ruido o error que afectan, respectivamente, al modelo de actuación del estado y al de medida. De acuerdo con las restricciones, estas señales de ruido se reescriben así (ecuación <2.12> del capítulo 2):

$$\begin{aligned} \bar{v}_{j,t-1} &= \text{norm}(0, V)_3 \\ \bar{o}_{j,t-1} &= \text{norm}(0, O) \end{aligned}, \quad <7.6>$$

donde V y O son las matrices de covarianza que caracterizan a las dos señales de ruido $\bar{v}_{j,t}$ y $\bar{o}_{j,t}$ gaussianos y de media nula.

V y O son, por tanto, dos matrices diagonales de tamaño 3x3 en las que el valor de las componentes de la diagonal principal ($\sigma_{v,i} / i = \{x, y, z\}$ para V , y $\sigma_{o,i} / i = \{x, y, z\}$ para O) se ajusta empíricamente al obtenido para la ejecución del XPFCP (ver el apartado 6.1.3 en el capítulo 6): $\sigma_{v,i} = 100mm$, y $\sigma_{o,i} = 150mm$. El correcto ajuste de estos parámetros así como la sensibilidad del KF a su valor (un ajuste erróneo de V y O puede producir la degeneración del filtro –[Maybeck79]–) se analiza en el apartado siguiente.

³ Se ha eliminado el subíndice j de las matrices de covarianza V y O puesto que se supone el mismo modelo de ruido para todos los objetos implicados en la tarea de MTT de interés, como se explica en párrafos anteriores.

El modelo de estimación utilizado para la tarea de MTT de interés cumple, por tanto, las restricciones impuestas por la definición básica del KF. Sin embargo, conviene remarcar que la propuesta de estimación multimodal planteada en la tesis, al estar basada en un PF, presenta una flexibilidad que el KF no tiene, y que asegura, en el XPFCP, su convergencia bajo condiciones de estimación muy diversas (distintos tipos de sensores o modelos de movimiento). El JPDAF no es, por tanto, una solución adecuada a la tarea de MTT de interés en esta tesis al incumplir la especificación de flexibilidad planteada en el capítulo 1.

A partir del modelo planteado, el JPDAF resuelve el problema de MTT a través de las etapas de predicción y corrección del KF (ver ecuaciones del capítulo 2), que se modifican para incorporar la incertidumbre de asociación conjunta generada por el JPDA. Más concretamente, el valor corregido del estado $\bar{x}_{j,t}$ de cada objeto se obtiene, en cada uno de los $j=1:k_{out,t}$ KFs implementados, a través de las probabilidades conjuntas de asociación $p_{i,j,t}$ de cada hipótesis de seguimiento con todo el conjunto de medidas $Y_t = \{\bar{y}_{i,t}\}_{i=1}^{m_t}$ obtenido en el cada instante⁴.

El funcionamiento del JPDAF queda así definido a través de las ecuaciones de predicción y corrección mostradas a continuación:

1. *Etapas de predicción.* Cada filtro $j=1:k_{out,t-1}$ genera un valor predicho $\bar{x}_{j,t|t-1}$ del vector de estado de la estimación de correspondiente (a partir de la ecuación de estado de <7.5>), y de la matriz de covarianza del error de esa estimación $P_{j,t|t-1}$ (a partir de la solución óptima del KF, mostrada en la expresión <2.13> del capítulo 2 y especificada para el modelo \widehat{CV}):

$$\begin{aligned} \bar{x}_{j,t|t-1} &= \bar{x}_{j,t-1} + B \cdot \bar{u}_{j,t-1} + \bar{v}_{j,t-1} \\ P_{j,t|t-1} &= P_{j,t-1} + V \end{aligned}, \quad <7.7>$$

donde las componentes del vector $\bar{u}_{j,t-1} = [vx_{j,t-1} \quad vz_{j,t-1}]^T$ se obtienen a través de la expresión <5.15> del capítulo 5, aplicadas al modelo \widehat{CV} del objeto j de interés:

$$\begin{aligned} vx_{j,t-1} &= (x_{j,t-1} - x_{j,t-2})/t_s \\ vz_{j,t-1} &= (z_{j,t-1} - z_{j,t-2})/t_s \end{aligned} \quad <7.8>$$

2. *Etapas de corrección.* Cada filtro $j=1:k_{out,t}$ genera un valor final del vector de estado $\bar{x}_{j,t}$ de la hipótesis correspondiente, corrigiendo el predicho $\bar{x}_{j,t|t-1}$ con cada medida $\bar{y}_{i,t} / i=1:m_t$ y en función de la probabilidad de asociación conjunta correspondiente $p_{i,j,t}$, a través de la ecuación <2.28> del capítulo 2, modificada de acuerdo con el modelo \widehat{CV} :

$$\begin{aligned} \bar{x}_{j,t} &= \bar{x}_{j,t|t-1} + K_{j,t} \cdot \bar{r}_{j,t} \\ \bar{r}_{j,t} &= \sum_{i=1}^{m_t} (p_{i,j,t} \cdot (\bar{y}_{i,t} - \bar{y}_{j,t|t-1})) \end{aligned}, \quad <7.9>$$

⁴ Aplicando la simplificación del JPDA que considera únicamente los datos a asociar de una ventana temporal de un solo instante (ver en el capítulo 2).

donde $\vec{r}_{j,t}$ es el residuo o innovación conjunta o combinada de la hipótesis j y se obtiene, como muestra la expresión <7.9>, combinando los m_t residuos individuales ($\vec{y}_{i,t} - \vec{y}_{j,t|t-1}$) mediante la probabilidad conjunta correspondiente $p_{t,i,j}$. Cada residuo individual $i=1:m_t$ se obtiene de la diferencia entre cada medida $\vec{y}_{i,t}$ y el valor esperado de esa medida $\vec{y}_{j,t|t-1} = \vec{x}_{j,t|t-1}$ (según el modelo \widehat{CV} mostrado en la ecuación <7.5>) para la hipótesis j .

Para obtener el valor corregido del vector de estado $\vec{x}_{j,t}$ en esta etapa es necesario calcular la matriz de Kalman $K_{j,t}$ para la hipótesis j correspondiente; a partir de la cuál se actualiza, además, el valor de la matriz de covarianza del error de esta estimación $P_{j,t}$. Estos cálculos se realizan a través de las ecuaciones siguientes (obtenidas al especificar la expresión <2.14> del capítulo 2 para el modelo \widehat{CV} aquí usado):

$$\begin{aligned} K_{j,t} &= P_{j,t|t-1} (P_{j,t|t-1} + O)^{-1} \\ P_{j,t} &= (I - K_{j,t}) \cdot P_{j,t|t-1} \end{aligned} \quad <7.10>$$

Sólo resta definir la forma de cálculo de los valores de probabilidad de asociación conjunta $p_{i,j,t} / i=1:m_t, j=1:k_{out,t}$, que dependen de los de verosimilitud disjunta de cada asociación $\Lambda_{i,j,t}$. Estos últimos se obtienen a través de una variable de distancia ([Bar-Shalom88]), que en este caso se define euclídea para utilizar la misma forma de distancia empleada en el proceso de asociación incluido en el XPFCP (ver la ecuación <6.16> del capítulo 6). Los valores de verosimilitud disjunta se obtienen, por tanto, a partir de los residuos individuales, definidos en un párrafo anterior:

$$\begin{aligned} \Lambda_{i,j,t} &= e^{-d_{i,j,t}^2 / 2 \cdot O} / i=1:m_t, j=1:k_{out,t} \quad ^5 \\ d_{i,j,t}^2 &= (\vec{y}_{i,t} - \vec{y}_{j,t|t-1})^2 \end{aligned} \quad <7.11>$$

A partir de estos valores de verosimilitud disjunta $\Lambda_{i,j,t} / i=1:m_t, j=1:k_{out,t}$, los de probabilidad conjunta $p_{i,j,t}$ se obtienen a partir de la combinación matemática de todas las soluciones de asociación posibles entre los $k_{out,t}$ targets y las m_t medidas, mediante un proceso de cálculo difícilmente realizable en tiempo real para un número alto de targets $k_{out,t}$.

Debido a ello, en las aplicaciones prácticas del JPDAF a tareas de MTT se suelen incluir un proceso de gating para cada hipótesis de seguimiento $j=1:k_{out,t}$ (ver en el capítulo 2). Esta técnica restringe las medidas que es posible asociar a cada target $\vec{y}_{i,j,t} / i=1:m_{j,t}$ a las ubicadas en un entorno marcado por la distancia $distG$ ([Vermaak05a]) alrededor de la proyección en el

⁵ En [Bar-Shalom88] esta expresión incluye una ponderación que desaparece normalizar el resultado de la ecuación <7.13> posterior, por lo que se elimina aquí para simplificar el cálculo.

espacio de medida, de la posición estimada para cada hipótesis j , en el paso de predicción del KF correspondiente: $\bar{y}_{j,t|t-1} = \bar{x}_{j,t|t-1}$ ⁶.

Aplicando esta etapa de gating al proceso de asociación conjunta, el valor de cada $p_{i,j,t} / i=1:m_{j,t}, j=1:k_{out,t}$ se obtiene a través de la siguiente ecuación ([Bar-Shalom88]):

$$p_{i,j,t} = \sum_{\Phi_{i,j,t}} \left(\prod_{a=1}^{m_t} (\Lambda_{a,t})^{\delta 1_{a,t}} \cdot \prod_{a=1}^{k_{t,out}} \left((p_D)^{\delta 2_{a,t}} \cdot (1-p_D)^{\delta 2_{a,t}} \right) \right), \quad <7.12>$$

donde $\Phi_{i,j,t}$ es el conjunto de soluciones de asociación que, tras el proceso de gating, incluyen la asociación entre el target j y la medida i ; p_D es la verosimilitud a priori de cada target $j=1:k_{t,out}$, supuesta idéntica para todos ellos; y $\delta 1_{a,t} / a=1:m_t$ (indicador de medida asociada) es una variables booleana que toma valor no nulo si la medida l está asociada a algún objeto $j=1:k_{t,out}$ y valor nulo en caso contrario; y $\delta 2_{a,t} / a=1:k_{t,out}$ (detector de target) es otra variables booleana que toma valor no nulo si alguna medida $i=1:m_t$ está asociada al objeto l , y valor nulo en caso contrario.

La expresión <7.12> sigue siendo, sin embargo, difícil de resolver en tiempo real para un valor de $k_{out,t}$ elevado. Además, en la definición del JPDA básico se especifica que sólo se puede asociar una medida con cada target ([Bar-Shalom88]), por lo que no puede aplicarse directamente a las pruebas resueltas con éxito en el capítulo 6 por el XPFCP. Existen múltiples propuestas de modificación del JPDA en la literatura del área que disminuyen el problema computacional, eliminando además la limitación comentada:

- En la mayor parte de aplicaciones del JPDA al tracking con visión se propone agrupar las medidas de Y_t en clases o blobs (p. ej. en [Isard01] o en [Vermaak05b]), para usarlas de este modo en el proceso de MTT.
- En muchos trabajos se plantean métodos matemáticos para obtener un valor aproximado al generado con la ecuación <7.12> para $p_{i,j,t} / i=1:m_{j,t}, j=1:k_{out,t}$: p. ej. un muestreador de Gibbs en [Hue02], o un PF en [Schulz03a].
- En otros trabajos se obtiene una modificación de la expresión <7.12> para obtener de forma más fácil el valor de la probabilidad conjunta $p_{i,j,t} / i=1:m_{j,t}, j=1:k_{out,t}$: en [Roecker93] se plantea el “JPDA Subóptimo” (“Suboptimal JPDA”), y en [Bar-Shalom90] el “JPDA Barato” (“Cheap JPDA”) propuesto por R.J. Fitzgerald.

⁶ En otras referencias como [Bar-Shalom88] el proceso de eliminación de hipótesis por gating se define a partir de un valor mínimo de verosimilitud p_G (si $\Lambda_{i,j,t} < p_G$). Este valor se obtiene sustituyendo $distG$ en la ecuación <7.12>. Se decide definir el gate del JPDAF en el espacio de distancias para destacar la equivalencia de este proceso con el de gating incluido en el clasificador de medidas descrito en el capítulo 5.

El JPDAF desarrollado en esta tesis para ser comparado funcionalmente con el XPFCP propuesto está basado en el uso de un “JPDA Barato”. En este algoritmo, las probabilidades conjuntas $p_{i,j,t} / i=1:m_{j,t}, j=1:k_{out,t}$ se obtienen a través de la siguiente expresión:

$$p_{i,j,t} = \frac{\Lambda_{i,j,t}}{\psi_{j,t} + \psi_{i,t} - \Lambda_{i,j,t} + \Lambda_0}, \quad <7.13>$$

donde Λ_0 es un factor de offset que puede anularse si el número de outliers en el proceso de asociación no es demasiado elevado; $\psi_{j,t} / j=1:k_{out,t}$ es la probabilidad global del target j , definida en la ecuación <7.14> siguiente; y de forma equivalente $\psi_{i,t} / i=1:m_t$ es la verosimilitud global de la medida i , definida por la expresión <7.15>:

$$\psi_{j,t} = \sum_{i=1}^{m_{j,t}} \Lambda_{i,j,t}, \quad <7.14>$$

$$\psi_{i,t} = \sum_{j=1}^{k_{out,t}} \Lambda_{i,j,t}, \quad <7.15>$$

En las expresiones <7.14> y <7.15> el efecto del proceso de gating se refleja en el cálculo de $\psi_{j,t}$ pero no en el de $\psi_{i,t}$, donde se tiene en cuenta al incluir en el sumatorio sólo aquellos targets $j=1:k_{out,t}$ que encierran en su gate a la medida i .

Esta implementación del JPDA genera a través de la ecuación <7.13> una aproximación bastante cercana al valor de $p_{i,j,t} / i=1:m_{j,t}, j=1:k_{out,t}$ que se obtiene con la ecuación <7.12>, y se ejecuta mucho más rápidamente que ésta. La expresión <7.13> es incluso válida cuando hay varias medidas asociadas al mismo objeto (se elimina así la restricción impuesta por el JPDA básico), por lo que es especialmente interesante en la aplicación de interés en la tesis.

La funcionalidad del JPDAF queda así completamente definida para su aplicación a la tarea de MTT para un número constante de targets, como especifica la versión básica del algoritmo. Para usar este algoritmo de seguimiento multi-objeto en la aplicación de interés es necesario, por tanto, establecer un mecanismo de creación y eliminación de hipótesis, y por ende, de KFs.

Como se expone en el capítulo 2, uno de los métodos más comúnmente empleados para que el JPDAF sea aplicable en situaciones con un número variable de hipótesis se basa en el uso del proceso de gating ya incluido en el algoritmo de MTT, de la siguiente forma:

1. Se crea una nueva hipótesis de estimación, $k_{out,t} = k_{out,t-1} + 1$ (se inicializa un nuevo KF) por cada medida $\vec{y}_{i,t} / i=1:m_t$ fuera del gate de todas las $k_{out,t}$ hipótesis existentes. El vector de estado de cada nuevo KF se inicializa en función de la medida no asociada ($\vec{x}_{k_{out,t},t} = \vec{y}_{i,t}$ según el modelo \mathbb{CV} usado), y la matriz de error de estimación $P_{k_{out,t},t}$ con un valor igual al de la matriz de covarianza del ruido de estado V .

2. Se elimina la hipótesis de estimación j , $k_{out,t} = k_{out,t-1} - 1$ (se elimina el KF usado para estimar $\bar{x}_{j,t}$) si ninguna medida se ubica dentro de su gate ($m_{j,t} = 0$).

Este método de creación y eliminación de hipótesis es muy sensible a variaciones instantáneas de la distribución de medidas entre los objetos implicados en el tracking, muy habituales en el sistema de observación usado para probar el comportamiento del XPFCP en esta tesis. El JPDAF así definido tiene, por tanto, una eficiencia muy baja, por lo que en algunos trabajos (p. ej. [Ng06]) se establece una ventana de validación e invalidación de las hipótesis, semejante al proceso de validación incluido en el clasificador propuesto en esta tesis.

El algoritmo JPDAF usado para realizar la comparativa perseguida se completa, por tanto, con esta modificación, y su funcionalidad global se muestra en la Figura 7.1. En esta figura, las etapas del núcleo de estimación del JPDAF (el KF) se muestran en color naranja, la de su núcleo de asociación JPDA en malva, y los pasos del proceso de modificación del número de hipótesis ($k_{out,t}$) añadido (basado en el gating) en verde.

En su aplicación a la tarea de MTT de interés, cada KF del JPDAF genera como estimación del estado de cada objeto $j = 1:k_{out,t}$ una gaussiana $norm(\bar{x}_{j,t}, P_{j,t})$, de media $\bar{x}_{j,t}$ (obtenidas a través de la expresión <7.9>) y varianza $P_{j,t}$ (dada por la ecuación <7.10>), indicadora de la fiabilidad de la estimación obtenida.

El JPDAF no cumple, por tanto, los objetivos planteados en la tesis para la tarea de tracking multi-objeto, debido a las razones que se enumeran a continuación:

- Implementa un estimador (un KF) por cada objeto, por lo que su carga computacional, y por ende su t_{exe} , depende linealmente de $k_{out,t}$.
- Incluye un algoritmo de asociación computacionalmente caro (el JPDAF) que impide su ejecución en tiempo real para aplicaciones de más de tres o cuatro objetos (como se demuestra en el apartado siguiente).
- No es flexible a la modificación del modelo de estimación usado (la convergencia de los KFs, y por tanto del JPDAF, sólo se asegura para modelos probabilísticos lineales con unas especificaciones de ruido muy concretas –[Maybeck79]–).

En el apartado siguiente se incluye el análisis empírico que completa y justifica la comparación teórica expuesta en los párrafos anteriores, entre esta propuesta, típicamente usada en aplicaciones de MTT ([Bar-Shalom88], [Rasmussen01] o [Vernaak05b]), y la planteada en esta tesis y descrita en el capítulo 6: el XPFCP.

7.2.2. Resultados comparativos del JPDAF frente al XPFCP

En este apartado se muestran los resultados generados por el JPDAF en los mismos experimentos y términos de fiabilidad, robustez y t_{exe} usados para validar la funcionalidad del

XPFCEP en la tarea de MTT de interés, en el capítulo 6. Al ser obtenidos bajo las mismas condiciones de test, los resultados así conseguidos permiten realizar la comparación empírica entre JPDAF y el XPFCEP que complementa a la teórica expuesta en el punto anterior.

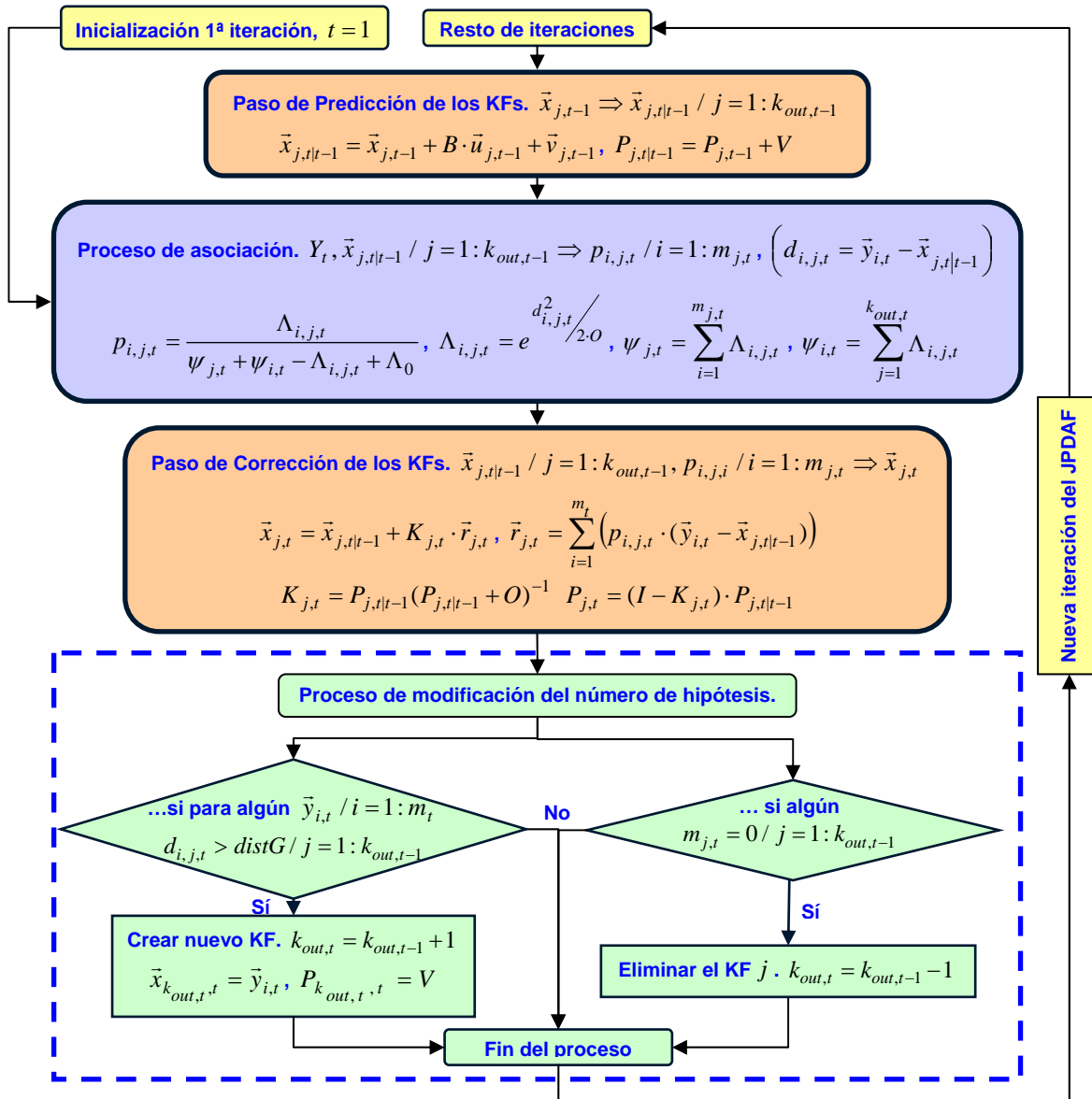


Figura 7.1. Diagrama funcional del algoritmo JPDAF aplicado a la tarea de MTT de interés. Las etapas del núcleo de estimación (los KFs) se muestran en color naranja, la del algoritmo de asociación JPDA en malva, y los pasos del proceso de modificación del número de hipótesis, basado en el gating, en verde.

Los resultados mostrados en los puntos siguientes se comparan, por tanto, con los expuestos en los apartados 6.1 y 6.2 del capítulo 6, para concluir el mejor comportamiento del JPDAF o del XPFCEP, en función de cada una de las características analizada.

a) Sensibilidad y ajuste de los parámetros del JPDAF

En primer lugar, se analiza la sensibilidad del JPDAF frente a la modificación de sus parámetros: las matrices V y O , que definen el modelo de ruido de estado y medida usado; y el parámetro $distG$ usado para realizar el gating.

La Figura 7.2 y la Figura 7.3 muestran la tasa de error generada por el JPDAF en función del valor asignado a la desviación típica que define las componentes de la diagonal principal de las matrices V ($\sigma_{v,i}/i=\{x,y,z\}$) y O ($\sigma_{o,i}/i=\{x,y,z\}$), respectivamente. Los resultados se obtienen en dos experimentos distintos (en azul los del test complejo, en verde los del completo corto y en rojo la media de ambos) y para el mismo rango de valores de $\sigma_{v,i}$ y $\sigma_{o,i}$ usado en el estudio equivalente realizado sobre el XPFCP y mostrado en el capítulo 6 (apartado 6.1.3).

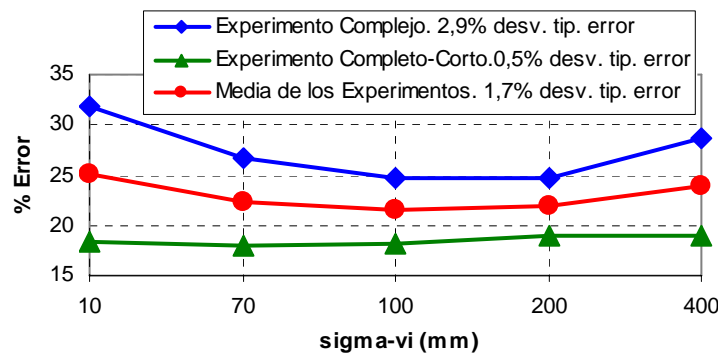


Figura 7.2. Tasa de error generada por el JPDAF en función de la desviación típica ($\sigma_{v,i}$, sigma-vi) de las componentes del vector de ruido de estado \vec{v}_t , supuestas gaussianas, incorreladas y de media nula, en el experimento complejo (en azul) y en el completo corto (en verde), y valor medio de ambos resultados (en rojo). Esta figura es equivalente a la Figura 6.19 del capítulo 6, que muestra la sensibilidad del XPFCP en estas condiciones.

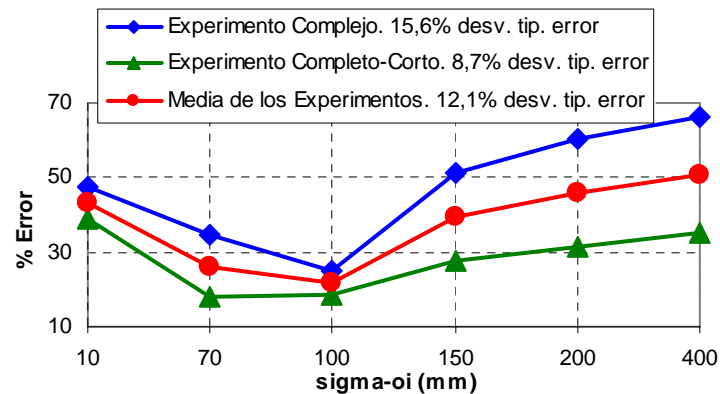


Figura 7.3. Tasa de error generada por el JPDAF en función de la desviación típica ($\sigma_{o,i}$, sigma-oi) de las componentes del ruido de medida \vec{o}_t , en los tests mostrados en la Figura 7.2. Esta figura es equivalente a la Figura 6.21 del capítulo 6, que muestra la sensibilidad del XPFCP en estas condiciones.

Del estudio de las Figuras Figura 7.2 y Figura 7.3 y sus equivalentes en el análisis realizado sobre el XPFCP (Figuras 6.19 y 6.21 del capítulo 6), se extraen las conclusiones listadas a continuación:

- Los dos algoritmos de tracking (JPDAF y XPFCP) son más sensibles a la modificación del ruido de medida $\bar{\sigma}_t$ que a la del ruido de estado \bar{v}_t (la desviación típica del error, en el mismo margen de modificación de los parámetros, es entre 7 y 8 veces mayor en los resultados obtenidos al modificar $\sigma_{o,i}$ que en los obtenidos al cambiar $\sigma_{v,i}$). Conviene, por tanto, analizar en mayor detalle la sensibilidad de los algoritmos frente a modificaciones de $\sigma_{o,i}$ que frente a cambios de $\sigma_{v,i}$.
- A la vista de las Figuras Figura 7.2 y Figura 7.3, el XPFCP parece ser incluso más sensible que el JPDAF ante cambios de cualquiera de los dos parámetros de varianza $\sigma_{o,i}^2$ o $\sigma_{v,i}^2$ (en ambos casos la desviación típica del error es, de media, doble en los resultados generados por el XPFCP que en los obtenidos con el JPDAF). Sin embargo, analizando en detalle la sensibilidad del XPFCP frente al parámetro más problemático ($\sigma_{o,i}^2$, según se concluye en el párrafo anterior) en la Figura 6.21 del capítulo 6 se observa que los cambios más importantes en la tasa de error generada por éste se producen para cambios grandes de $\sigma_{o,i}^2$. Una modificación de este parámetro más acorde (del 50% respecto a su valor de ajuste final en el XPFCP, es decir $\sigma_{o,i} = [75,225]mm$) con los valores que presenta en los sets de datos usados ($\sigma_{o,i} = [40,230]mm$ según se indica en el apéndice IV) indica que la desviación típica media de la tasa de error generada por el XPFCP alcanza sólo el 5.6% frente al 11.4% obtenido del JPDAF en estas nuevas condiciones de prueba.
- El valor óptimo de ajuste de $\sigma_{v,i}$ en los experimentos mostrados resulta ser idéntico para ambos algoritmos $\sigma_{v,i|opt} = 100mm^7$, mientras que el de $\sigma_{o,i}$ no coincide: en el XPFCP $\sigma_{o,i|opt} = 150mm$ y en el JPDAF $\sigma_{o,i|opt} = 100mm$ (de la Figura 7.3).

Los análisis expuestos indican que, desde el punto de vista del modelo de ruido, ambas propuestas de seguimiento (XPFCP y JPDAF) tienen una sensibilidad semejante. La mayor flexibilidad del XPFCP frente al JPDAF se presenta, sin embargo, en la forma del modelo, como se explica en el apartado anterior.

Por otra parte, la Figura 7.4 muestra la tasa de error generada por el JPDAF en el experimento complejo, en función del valor asignado al parámetro de gating $distG$. Este parámetro es equivalente al de gating del clasificador de medidas $Y_t \Rightarrow G_{1:k,t|in}$, $distM$, ya que este segmentador actúa como proceso de asociación de tipo NN en el XPFCP. La Figura 7.4 es, por tanto, equivalente, a la Figura 5.23 del capítulo 5, y para facilitar la comparación de estas figuras el valor de este parámetro se referencia en la Figura 7.4 como $distG$ (en la Figura 5.23 del capítulo 5 este valor se muestra como $distM$), si bien tiene un valor igual al de ajuste óptimo de $distM_{opt} = distG = 640mm$ (ver en el apartado 6.1.3).

⁷ El estudio aquí incluido sobre el ajuste de $\sigma_{v,i}$ se hace en términos de posición, como el equivalente realizado sobre el XPFCP, aunque la magnitud de esta variable no es igual para todas las componentes $i = \{x, y, z, vx, vz\}$.

Del análisis de los resultados mostrados en la Figura 7.4 y su comparación con los expuestos en la Figura 5.23 del capítulo 5 se extraen las siguientes conclusiones:

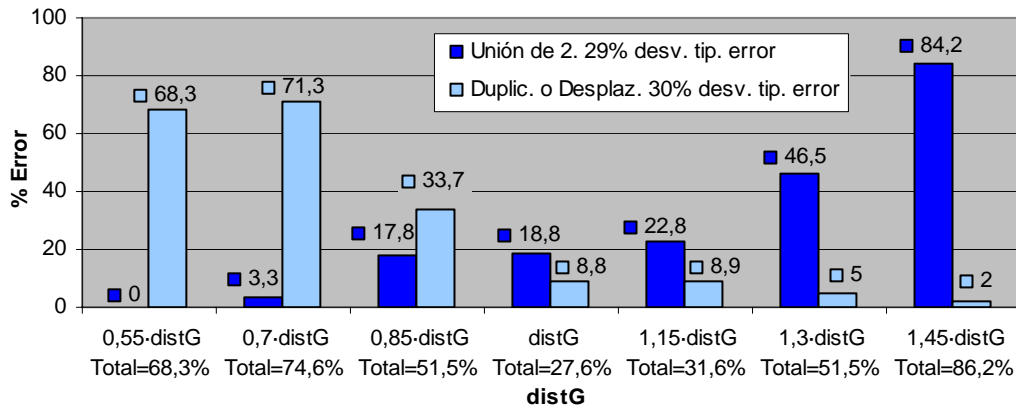


Figura 7.4. Tasa de error generada por el JPDAF en función del valor asignado al parámetro de gating $distG$ en el experimento complejo. Esta figura es equivalente a la Figura 5.23 del capítulo 5 que muestra la sensibilidad, en estas mismas condiciones, del clasificador que actúa como proceso de asociación en el XPFCP.

- El JPDAF es muy sensible a la modificación de $distG$. Mientras que la tasa de error del XPFCP es prácticamente constante al modificar $distM$ en hasta un 30% de su valor de ajuste (gracias a los procesos de predicción de centroides y de validación incluidos en el clasificador que realiza la asociación en el XPFCP, como se explica en el capítulo 5), la tasa generada por el JPDAF presenta una desviación típica media del 21.8% (un 15.6% en los errores por unión de 2 hipótesis en 1 y un 28% en los de duplicación o desplazamiento de hipótesis) ante una modificación de $distG$ de esa misma magnitud.
- El valor de ajuste óptimo para el parámetro $distG$ resulta ser igual al usado para el parámetro $distM$.

La funcionalidad del JPDAF es, por tanto, muy sensible a modificaciones del parámetro de gating $distG$: si éste tiene un valor menor al de ajuste óptimo se incrementa en gran medida la duplicación de hipótesis de seguimiento (y por tanto, el número de KFs implicados en el JPDAF y con él, el tiempo de cómputo del seguidor); si, por el contrario, se asigna a $distG$ un valor mayor al de ajuste óptimo aumenta el número de errores de unión de 2 hipótesis en 1, es decir, de objetos no seguidos de forma independiente por el JPDAF. Este comportamiento es el mismo que presenta el clasificador “*k-medias secuencial*” sin validación ni proceso de predicción de centroides (el “*k-medias extendido*”), mostrado en la Figura 5.9 del capítulo 5.

El análisis expuesto confirma que las mejoras incluidas en el clasificador que realiza la tarea de asociación en el XPFCP insensibilizan el funcionamiento del seguidor multi-objeto del valor de este parámetro de gating, aportándole una robustez que no tiene el JPDAF típicamente usado, sin embargo, en aplicaciones de MTT.

b) Funcionalidad, fiabilidad y robustez frente al ruido

Realizado el ajuste de los parámetros del JPDAF se procede a comparar la fiabilidad de este algoritmo frente a la del XPFCP. Para ello en la Figura 7.5 se muestra la tasa de error generada

por ambos algoritmos (en verde el S-JPDAF y en azul el XPFCP) en los experimentos complejo (a), completo corto (b) y de ruido en función del tipo de error. Las conclusiones extraídas de esta figura sobre la fiabilidad comparativa se listan a continuación:

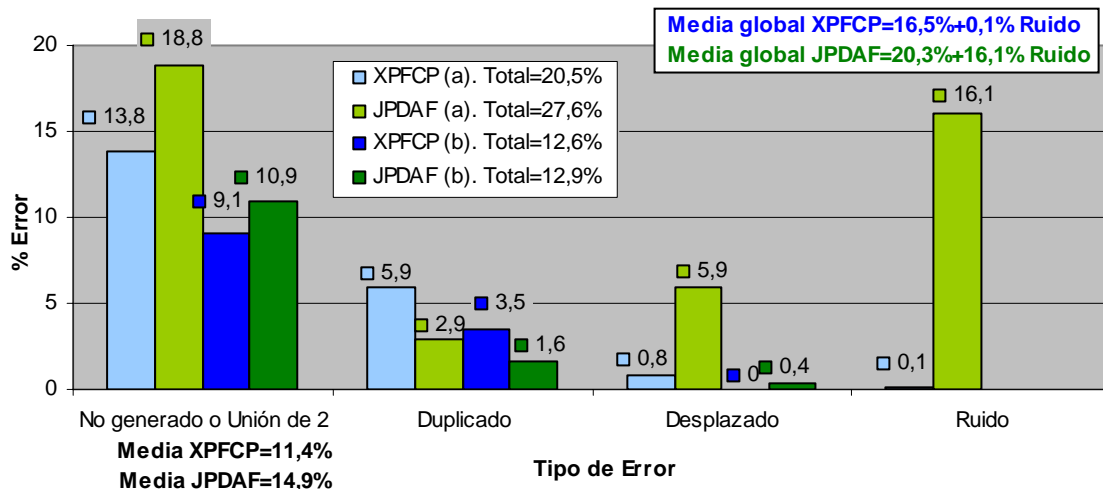


Figura 7.5. Tasas de error generadas por el JPDAF (en verde) y el XPFCP (en azul) en función del tipo de error, en los tests complejo (tonos claros), completo corto (tonos oscuros) y de ruido (para la tasa de ruido no filtrado).

- El JPDAF presenta menor fiabilidad que el XPFCP (genera una tasa de error 7 puntos mayor) en situaciones difíciles de tracking como las presentes en el test complejo. En situaciones de complejidad diversa (como las que se dan en el test completo corto) ambos algoritmos presentan, sin embargo, una fiabilidad semejante (ambos generan una tasa de error algo menor del 13%).
- En ambos experimentos, el JPDAF genera más errores de unión de 2 hipótesis en 1 que el XPFCP (una media de un 3.5% más de este tipo de error, que es además el más importante en esta aplicación, como ya se demuestra en el apéndice IV). El XPFCP genera, sin embargo de media un 2.5% más hipótesis duplicadas que el JPDAF, pero este tipo de error es menos relevante en la aplicación de MTT de interés, y además no tiene una repercusión importante en el t_{exe} del XPFCP, como se demuestra en el apartado 6.2.2. Finalmente, llama la atención la elevada tasa de error por desplazamiento de las hipótesis que genera el JPDAF en el test complejo (un 5.9%), y que es casi nula (un 0.8%) en los resultados arrojados por el XPFCP. En un punto posterior se demuestra que este último tipo de errores generado por el JPDAF se debe al mal funcionamiento de la asociación JPDA que este seguidor incluye, fundamentalmente en las situaciones de oclusión parcial o total de alguna de las hipótesis que se producen en los cruces (muy abundantes en este test complejo, como se muestra en el apéndice III). El XPFCP gestiona mejor este tipo de situaciones complejas de tracking gracias al proceso de validación de los clasificadores de medidas y partículas, como se demuestra en el apartado 6.2.2.
- Finalmente la tasa de ruido no filtrado arrojada por el JPDAF es mucho mayor que la generada por el XPFCP, que presenta un rechazo al ruido de casi el 100%, gracias de nuevo al proceso de validación incluido en los clasificadores de medidas y partículas. De hecho, el

JPDAF no incluye ningún mecanismo de rechazo al ruido, pero filtra gran parte de los outliers ya que no es capaz de completar la generación de un KF asociada a hipótesis como ésta que permanecen durante una sola iteración de proceso de seguimiento.

Las conclusiones aquí expuestas permiten asegurar que el XPFCP es más fiable que el JPDAF en situaciones complejas de seguimiento y presenta un mayor rechazo al ruido, si bien ambos algoritmos tienen buen comportamiento en contextos sencillos de tracking.

Para completar el análisis de fiabilidad expuesto, en la Figura 7.6 se muestran los resultados obtenidos en la ejecución real en la segunda parte del experimento complejo de los dos algoritmos de MTT comparados (JPDAF en la imagen derecha y XPFCP en la izquierda). En esta figura, equivalente a la Figura 6.37 del capítulo 6, se representa por colores la posición estimada por cada algoritmo para cada uno de los cuatro objetos implicados en esta prueba de tracking (dada por $\bar{x}_{1:5}$ en el caso del JPDAF y por $\bar{g}_{1:4|out}$ en el caso del XPFCP). Sobre la secuencia de puntos de color se indica con una flecha, en esta Figura 7.6 como en la mencionada Figura 6.37, el “ground truth” manual de posición y dirección del movimiento de los objetos en cada instante del experimento con el que comparan los resultados arrojados por los seguidores. Las conclusiones extraídas de la Figura 7.6 son las siguientes:

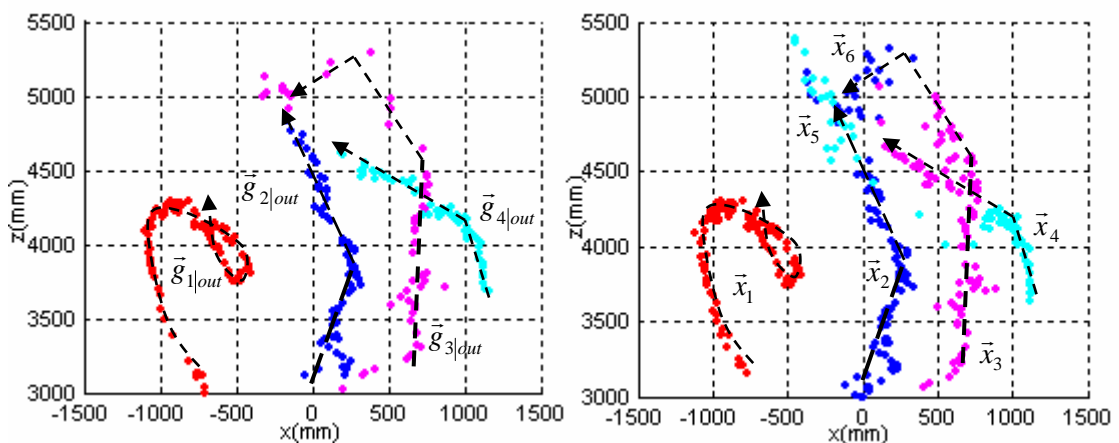


Figura 7.6. Representación por colores de la posición en el plano XZ estimada en cada instante por el XPFCP (marcada como $\bar{g}_{j|out}$ en la imagen de la izquierda) y por el JPDAF (marcada como \bar{x}_j en imagen de la derecha) de los cuatro objetos implicados en el proceso de tracking en la segunda parte del experimento complejo.

- El JPDAF, al no incorporar un proceso explícito o implícito de identificación (como el que sí incluye el XPFCP a través del clasificador de medidas) elimina al objeto identificado como 4 en el cruce que se produce entre éste y el identificado como 3 (el número 4 pasa antes por el punto de cruce que el 3, por lo que este último oculta al primero). El proceso de asociación JPDA no crea una identificación explícita de los targets y no incorpora un proceso de validación como el que sí tiene el clasificador de medidas incluido en el XPFCP, por lo que durante la situación de oclusión, el KF asociado al target 4 se elimina, y tras la oclusión, las medidas que se obtienen de éste y el XPFCP asocia correctamente a éste son, sin embargo asociadas al target 3 ya que el 4 no existe. Esta situación da lugar, por tanto, a un error del tipo unión de 2 hipótesis en 1, habitual en el JPDAF, como demuestra la Figura 7.5.

- Al final del experimento vuelve a repetirse un problema similar al descrito en el párrafo anterior con los objetos numerados como 2 y 3. Estos son eliminados para dar lugar a dos nuevos objetos (numerados como 5 y 6 aunque identificados con los colores azul claro y oscuro ya usados) debido a que las medidas relacionadas con ellos no se ubican dentro del gate $distG$ correspondiente (evidenciándose la sensibilidad del JPDAF respecto a $distG$).

Queda así demostrado que el XPFCP aporta al proceso de MTT una funcionalidad de identificación de objetos (gracias las características del algoritmo de clasificación de medidas diseñado para realizar la tarea de asociación implícita en el seguimiento) que no proporciona el JPDAF y que lleva a este último algoritmo a generar errores de seguimiento.

c) Tiempo de ejecución

Otra característica a analizar en la comparativa entre el JPDAF y el XPFCP es el coste computacional, y con él, el t_{exe} de ambos algoritmos.

La Figura 7.7 muestra el t_{exe} ⁸ de ambos algoritmos (en verde el del JPDAF y en azul el del XPFCP) en el experimento complejo. La figura demuestra que este tiempo es siempre inferior (una media de casi 4 veces en este test) en el XPFCP que en el JPDAF.

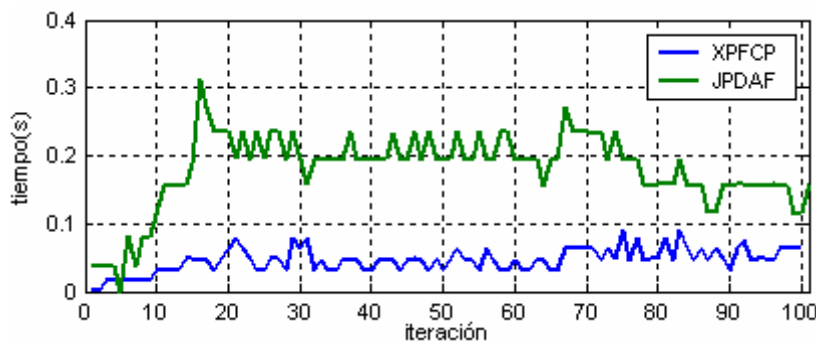


Figura 7.7. Gráficas comparativas del t_{exe} del XPFCP (en azul) y del JPDAF (en verde) en el test complejo.

Existen dos razones principales, ya comentadas en el apartado anterior, que justifican este resultado: en primer lugar, el proceso de asociación incluido en el JPDAF (el JPDA, de complejidad computacional $O((k_{out,t} + 1)!^{m_t})$ -ver en el capítulo 2-) es computacionalmente más caro que el incluido en el XPFCP (un NN sobre las clases de medidas, de complejidad computacional $O(n^{k_{in,t}}/k_{in,t} \ll m_t)$); en segundo lugar, el JPDAF incluye como núcleo de estimación tantos KFs como hipótesis de seguimiento haya implicadas en la tarea de tracking, mientras que en el XPFCP este núcleo está formado por un solo PF multimodal.

La Figura 7.8 muestra el t_{exe} del JPDAF (en verde) y del XPFCP (en azul) en función del número de objetos implicados en el seguimiento ($k_{out,t}$) en el experimento completo largo, y

⁸ Como en el capítulo 6, las medidas comparativas de t_{exe} se han realizado sobre una plataforma de ejecución en tiempo real desarrollada en C, y corriendo en un procesador Intel Centrino DUO a 1.8GHz.

demuestra que esta relación es prácticamente exponencial en el caso del JPDAF, y sin embargo el t_{exe} del XPFCP es casi constante e independiente de $k_{out,t}$.

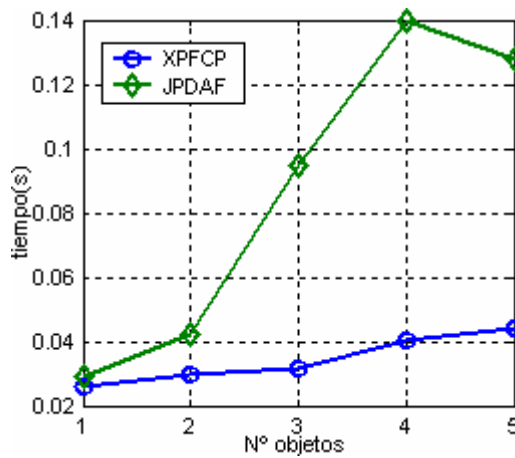


Figura 7.8. Gráficas comparativas del t_{exe} del XPFCP (en azul) y del JPDAF (en verde) en el test completo largo.

Los resultados expuestos corroboran que el XPFCP cumple con la especificación planteada en el capítulo 1 de que el t_{exe} del seguidor multi-objeto sea constante con $k_{out,t}$, facilitándose así su ejecución en tiempo real en cualquier situación. Por su parte, el JPDAF no cumple esta especificación (pues su t_{exe} cambia con $k_{out,t}$) y solo supera la velocidad de ejecución del XPFCP si el número de targets y de medidas⁹ es bajo.

La Tabla 7.1 muestra el consumo de tiempo (en el test complejo) que conllevan los procesos de asociación y estimación del JPDAF (tantos KFs como hipótesis y un JPDA) y en el XPFCP (un PF y un “*k-medias secuencial*” con validación), relativos al t_{exe} global de cada algoritmo.

Tabla 7.1. Valor medio absoluto y relativo del tiempo de cómputo de los procesos de asociación y estimación implicados en el JPDAF y el XPFCP al ser ejecutados en el experimento complejo.

Clasificador de medidas	t_{exe} total (s)	t_{exe} asociación (s, % del total)	t_{exe} de estimación (s, % del total)
XPFCP	0.045	0.013, 29%	0.032, 71% (36% del PF + 35% del cluster de partículas)
JPDAF	0.179	0.143, 79%	0.036, 21%

Los resultados expuestos en la Tabla 7.1 confirman que la mayor parte de la carga computacional del JPDAF la produce el proceso de asociación JPDA (un 79%) mientras que la ejecución de los KFs (3 filtros asociados a los 3 objetos que de media aparecen en el test complejo) solo conlleva el 21% de esta carga. Por el contrario, el proceso de asociación (clasificación determinística del set de medidas Y_t) del XPFCP solo produce el 29% de su carga computacional total, mientras que el 71% restante se reparte entre el lazo de ejecución del PF

⁹ En los experimentos puestos en práctica en este capítulo se usan una media de 175 medidas de posición en cada frame del sistema de seguimiento multi-objeto.

(que produce un 36% de esa carga total) y la clasificación del set de partículas a la salida del XPFCP (que produce un 35% de ese total).

Analizando de forma absoluta el t_{exe} de estimación en los dos algoritmos se observa que éste es prácticamente igual. Este t_{exe} implica la ejecución de una media de 3 KFs (12ms por KF) en el JPDAF, y la del PF modificado (16ms) y del clasificador de partículas (15ms) en el XPFCP. Se observa, por tanto, que si bien de media en todo el test complejo el t_{exe} del KF es algo menor al del PF, la diferencia en estos tiempos es, absoluta y relativamente, tan pequeña que compensa emplear un algoritmo flexible como el PF en lugar del KF (cuyo uso implica el conjunto de restricciones ya comentadas) como núcleo de estimación del seguidor.

Como conclusión de lo expuesto, en la Tabla 7.2 se muestra la velocidad media de ejecución de los dos algoritmos de seguimiento comparados en este apartado en los distintos experimentos probados. Los resultados expuestos en esta tabla confirman, una vez más, que el XPFCP se ejecuta a velocidad constante y en tiempo real (superior a 15FPS), independientemente de la complejidad de la situación de tracking, mientras que en el caso del JPDAF esta velocidad no es constante y disminuye en situaciones complejas (a), impidiendo su ejecución en tiempo real.

Tabla 7.2. Velocidad media de ejecución del XPFCP y del JPDAF en los experimentos complejo (a), completo corto (b) y completo largo (c)

Algoritmo/Test	FPS(a)	FPS(b)	FPS(c)
XPFCP	22	23.6	25
JPDAF	5.6	8	11

Los análisis de sensibilidad, robustez, fiabilidad, t_{exe} y funcionalidad expuestos, tienen como conclusión que el XPFCP presenta un mejor comportamiento que el JPDAF en la tarea de MTT de interés, y que al contrario que este último, el XPFCP cumple todas las especificaciones planteadas para esta tarea al principio de la tesis.

7.2.3. El “JPDAF Muestreado” (“Sampled JPDAF”, S-JPDAF)

Como se comenta en la introducción de este apartado, el otro algoritmo implementado en esta tesis para ser comparado a nivel teórico y empírico con el XPFCP propuesto es el S-JPDAF presentado en [Schulz01] y analizado en más detalle en [Schulz03b]. El S-JPDAF es una versión muestreada del JPDAF antes descrito, y surge, como se explica ya en el capítulo 2, de la incorporación del set de probabilidades conjuntas $p_{i,j,t} / i=1:m_{j,t}, j=1:k_{out,t}$ generadas por el JPDA (de funcionalidad descrita en el apartado 7.2.1) a la etapa de corrección o pesado de las partículas $\vec{x}_{j,t}^{(a)} / a=1:n^{10}$ de un PF.

¹⁰ En este apartado se cambia el subíndice i , que referencia generalmente a las muestras del set de partículas, por el subíndice a . El objetivo de este cambio es mantener la coherencia de la notación entre este apartado y el anterior. La letra i referencia en este apartado, al igual que en el anterior, a las medidas incluidas en el vector $Y_t = \{\vec{y}_{i,t}\}_{i=1}^{m_t}$.

El S-JPDAF aúna la fiabilidad del JPDA con la robustez y flexibilidad del PF, por lo que es repetidamente usado en distintas propuestas de MTT ([MacCormick99], [Isard01], [Schulz03b], [Vermaak05a], [Almeida05] o [Ng06]) y referenciado en diversos trabajos comparativos ([Karlsson01], [Hue02] o [Frank03]). Estas características del S-JPDAF, junto al hecho de que el algoritmo aporta una alternativa de asociación diferente a la incluida en el XPFCP propuesto, y tiene, sin embargo, el mismo núcleo de estimación que este último, llevan a elegir la versión muestreada del JPDAF para la comparativa perseguida.

En el capítulo 2 se expone que la versión del S-JPDAF basada en el uso de un sólo PF con vector de estado extendido (cada partícula representa la estimación conjunta del estado de todos los objetos $\chi_t^{(a)} = \left[\bar{x}_{1,t-1}^{(a)} \quad \bar{x}_{2,t}^{(a)} \quad \dots \quad \bar{x}_{k_{out,t}}^{(a)} \right]^T = \bigcup_{j=1}^{k_{out,t}} \bar{x}_{j,t}^{(a)} / a=1:n$) es la versión más usada de este algoritmo por la comunidad científica en aplicaciones de MTT ([Isard01], [Hue02] o [Ng06]). Sin embargo, debido al alto coste computacional del seguidor, ninguno de los trabajos listados muestra resultados de tracking en tiempo real para más de tres o cuatro objetos.

Debido a ello, en la comparativa aquí incluida se opta por usar la versión del S-JPDAF planteada en [Schulz03b], en la que se emplea un PF para estimar de forma individual el estado de cada objeto $\bar{x}_{j,t} / j=1:k_{out,t}$, y un único JPDA para obtener la probabilidad $p_{i,j,t} / i=1:m_{j,t}$ de las distintas hipótesis de asociación conjunta. El S-JPDAF así descrito se comporta como un conjunto de PFs casi independientes relacionados en su paso de corrección a través del proceso de asociación conjunta, de forma semejante al algoritmo MPF presentado en [Vermaak03] y analizado en el apartado 7.1.

En este caso se emplea el mismo modelo CV, representado por las expresiones de <6.18> del capítulo 6 que se repiten a continuación por claridad, para describir el comportamiento de cada target implicado en la tarea de MTT, empleando el mismo modelo para todos los objetos.

$$\begin{aligned} \bar{x}_t &= A \cdot \bar{x}_{t-1} + \bar{v}_{t-1} \\ \bar{y}_t &= C \cdot \bar{x}_t + \bar{o}_t \end{aligned} \quad \langle 7.16 \rangle$$

donde $A = \begin{bmatrix} 1 & 0 & 0 & t_s & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & t_s \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ es la matriz de transición del sistema que sólo depende del

periodo de muestreo del estimador discreto t_s , $C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ es la matriz de observación, y $\bar{v}_t = [v_{x,t} \quad v_{y,t} \quad v_{z,t} \quad v_{vx,t} \quad v_{vz,t}]^T$ y $\bar{o}_t = [o_{x,t} \quad o_{y,t} \quad o_{z,t}]^T$ son los vectores de ruido de estado y de medida respectivamente.

De este modo, el flujograma del S-JPDAF presenta máxima similitud con el del XPFCP propuesto. Teniendo en cuenta esta consideración, el esquema funcional del S-JPDAF se basa en las tres etapas típicas del PF básico:

1. *Etapas de predicción*, en la que se obtiene, a través del conjunto de partículas correspondiente, la representación discreta de la creencia a priori del estado de cada objeto implicado en el seguimiento $j = 1:k_{out,t-1}$: $S_{j,t|t-1} = \left\{ \bar{x}_{j,t|t-1}^{(a)}, \frac{1}{n} \right\}_{a=1}^n \cong p(\bar{x}_{j,t}^{(1:n)} | \bar{y}_{1:t-1})$. Para ello se aplica a las partículas del set de salida de cada PF en $t-1$, $S_{j,t-1}$, el modelo de actuación del objeto correspondiente $p(\bar{x}_{j,t} | \bar{x}_{j,t-1})$, que queda especificado en este caso, para todos los objetos, por la ecuación superior de la expresión <7.16>. La funcionalidad de esta etapa del S-JPDAF no se modifica, por tanto, respecto a la que tiene en el XPFCP.

$$p(\bar{x}_{j,t}^{(1:n)} | \bar{y}_{1:t-1}) = p(\bar{x}_{j,t-1}^{(1:n)} | \bar{y}_{1:t-1}) \cdot p(\bar{x}_{j,t} | \bar{x}_{j,t-1}) \quad <7.17>$$

$$\bar{x}_{j,t|t-1}^{(a)} = A \cdot \bar{x}_{j,t-1}^{(a)} + \bar{v}_{j,t-1}$$

donde $\bar{v}_{j,t-1}$ caracteriza al ruido que afecta al modelo de estado de cada objeto j , y que se define igual que el empleado en el XPFCP (al usar en el S-JPDAF el mismo modelo CV para caracterizar el comportamiento de los objetos implicados) como $norm(0, V)$: gaussiano, blanco, con media nula y matriz de covarianza diagonal, con un valor inicial (a falta del ajuste realizado en el apartado siguiente) de las componentes de la diagonal principal $\sigma_{v,i} = 100 / i = \{x, y, z, vx, vz\}$.

2. *Etapas de corrección*, en la que se actualizan los pesos $w_{j,t}^{(a)} / a = 1:n$ del set de partículas $S_{j,t|t-1}$ relacionado con cada objeto $j = 1:k_{out,t}$, en función de los valores de probabilidad conjunta $p_{i,j,a,t} / i = 1:m_{j,a,t}$ generados por un JPDA como el descrito en el apartado 7.2.1. En la función de ponderación de las partículas relacionada con cada objeto, $w(\bar{x}_{j,0:t})$, se usa como función de aproximación a la creencia la típica del algoritmo “*Bootstrap*” $q(\bar{x}_{j,t} | \bar{x}_{j,0:t-1}, \bar{y}_{1:t}) = p(\bar{x}_{j,t} | \bar{x}_{j,t-1})$, como se hace en el XPFCP y en la mayor parte de las aplicaciones prácticas del S-JPDAF encontradas en la literatura del área ([Schulz03b], [Frank03] o [Almeida05]). A partir de esta premisa, la función de pesado a usar en cada PF del S-JPDAF queda como se muestra:

$$w(\bar{x}_{j,0:t}) = w(\bar{x}_{j,0:t-1}) \cdot \frac{p(\bar{y}_t | \bar{x}_{j,t}) \cdot p(\bar{x}_{j,t} | \bar{x}_{j,t-1})}{q(\bar{x}_{j,t} | \bar{x}_{j,0:t-1}, \bar{y}_{1:t})} = w(\bar{x}_{j,0:t-1}) \cdot p(\bar{y}_t | \bar{x}_{j,t}), \quad <7.18>$$

donde la función de verosimilitud $p(\bar{y}_t | \bar{x}_{j,t})$ se especifica para cada partícula, siguiendo el mismo criterio de coherencia con propuestas como la presentada en [Schulz03b] o la planteada en esta tesis (ver la ecuación <6.11> del capítulo 6). La ecuación <7.18> queda así transformada como se muestra a continuación:

$$w_{j,t}^{(a)} = w_{j,t-1}^{(a)} \cdot p(\bar{y}_t | \bar{x}_{j,t}^{(a)}) \quad <7.19>$$

Tras este procedimiento, y como final de la etapa de corrección de cada PF en el S-JPDAF, se normalizan los sets de pesos, para dar lugar a los vectores $\vec{w}_{j,t} = [\tilde{w}_{j,t}^{(1:n)}] / j = 1:k_{out,t}$.

3. *Etapa de selección*, en la que se utiliza alguno de los algoritmos de resampling planteados en el capítulo 4 para obtener el set de salida de cada PF $j = 1:k_{out,t}$ del S-JPDAF $S_{j,t} = \{\vec{x}_{j,t}^{(a)}, 1/n\}_{a=1}^n$, a partir del conjunto de pesos $\vec{w}_{j,t} = [\tilde{w}_{j,t}^{(1:n)}]$ correspondiente. De nuevo por similitud con el XPFCP, en este paso del S-JPDAF se usa un algoritmo de resampling de tipo residual.

Como se explica en el apartado 7.2.1, la función de verosimilitud individual de cada partícula $p(\vec{y}_t | \vec{x}_{j,t}^{(a)})$ de la expresión <7.19> se obtiene con los valores de probabilidad conjunta $p_{i,j,a,t} / i = 1:m_{j,a,t}, a = 1:n, j = 1:k_{out,t}$ generados por el JPDA del S-JPDAF. De este modo, la funcionalidad del algoritmo de asociación se inserta en el lazo de ejecución de cada PF.

Más concretamente, aplicando el proceso de asociación conjunta a la función $p(\vec{y}_t | \vec{x}_{j,t}^{(a)}) / a = 1:n, j = 1:k_{out,t}$ y desarrollando su valor a través de la matemática probabilística correspondiente ([Bar-Shalom88] y [Schulz03b]) se obtiene el siguiente resultado:

$$p(\vec{y}_t | \vec{x}_{j,t}^{(a)}) = \sum_{i=1}^{m_{j,a,t}} \left(p_{i,j,a,t} \cdot e^{-d_{i,j,a,t}^2 / 2 \cdot O} \right), \quad <7.20>$$

donde $p_{i,j,a,t}$ es la probabilidad de asociación conjunta de la estimación de posición dada por la partícula a para el objeto j con la medida de posición i ; $d_{i,j,a,t}^2$ es una medida de distancia al cuadrado entre esos dos elementos; y O es la matriz de covarianza del ruido $\vec{o}_{j,t}$ incluido en el modelo de observación de cada objeto (ver ecuación inferior en la expresión <7.16>). Como valor inicial (a falta del ajuste realizado en el apartado siguiente) se asigna a la matriz O el empleado en el XPFCP (pues en el S-JPDAF se usa el mismo modelo CV que en aquel): matriz diagonal (supuesto $\vec{o}_{j,t}$ un ruido gaussiano, blanco y de media nula) con valor de sus componentes no nulas $\sigma_{o,i} = 150mm / i = \{x, y, z\}$.

La variable de distancia $d_{i,j,a,t} / i = 1:m_{j,a,t}, a = 1:n, j = 1:k_{out,t}$ se define euclídea, de nuevo para conseguir máxima similitud del flujograma del S-JPDAF con el del XPFCP.

De esta forma, la variable de distancia $d_{i,j,a,t} / i = 1:m_{j,a,t}, a = 1:n, j = 1:k_{out,t}$ queda definida del mismo modo que en el JPDAF descrito en el apartado 7.2.1 (ecuación <7.11>), sustituyendo el subíndice j por el conjunto a, j , que referencia a la estimación a obtenida por el filtro j para la hipótesis de seguimiento correspondiente:

$$d_{i,j,a,t}^2 = (\vec{y}_{i,t} - \vec{y}_{j,t|t-1}^{(a)})^2, \quad <7.21>$$

donde $\bar{y}_{j,t|t-1}^{(a)}$ es la medida de posición que se espera recibir del objeto j , a partir de la estimación a priori de su estado dada por la partícula a del filtro correspondiente $\bar{x}_{j,t|t-1}^{(a)}$. Este valor $\bar{y}_{j,t|t-1}^{(a)} = C \cdot \bar{x}_{j,t|t-1}^{(a)}$ se obtiene aplicando la ecuación de medida del modelo CV de cada objeto (expresión inferior de <7.16>) a la estimación a priori obtenida para todas las partículas de cada filtro en el paso de predicción del S-JPDAF como se explica en el apartado 7.2.1.

Por otro lado, el valor de la probabilidad conjunta de asociación $p_{i,j,a,t} / i=1:m_{j,a,t}, a=1:n, j=1:k_{out,t}$ se define también de la misma forma que en el JPDAF (ecuación <7.13>), realizando una idéntica sustitución del subíndice j por el conjunto a, j :

$$p_{i,j,a,t} = \frac{\Lambda_{i,j,a,t}}{\Psi_{j,a,t} + \Psi_{i,t} - \Lambda_{i,j,a,t} + \Lambda_0} / \Lambda_{i,j,a,t} = e^{d_{i,j,a,t}^2 / 2\sigma^2}, \quad <7.22>$$

donde $\Lambda_{i,j,a,t}$ (calculada como se observa a partir del valor de $d_{i,j,a,t}^2$, definido en la ecuación <7.22>) es la probabilidad disjunta de que la estimación de estado del objeto j , dada por la partícula $\bar{x}_{j,t|t-1}^{(a)}$ esté asociada con la medida $y_{i,t}$; y $\Psi_{j,a,t}, \Psi_{i,t}$ se definen del mismo modo que en el JPDA (ver ecuaciones <7.14> y <7.15>) a partir de los valores de $\Lambda_{i,j,a,t}$ relacionados con cada partícula a de cada filtro j .

Para calcular el valor de la verosimilitud global $\Psi_{i,t}$ de cada medida $i=1:m_t$ es necesario tener en cuenta las n representaciones del estado de cada objeto $j=1:k_{out,t}$ dadas por el PF correspondiente. Es necesario, por tanto, desarrollar la ecuación <7.15> como se muestra:

$$\Psi_{i,t} = \sum_{j=1}^{k_{out,t}} \Lambda_{i,j,t} = \sum_{j=1}^{k_{out,t}} \sum_{a=1}^n \Lambda_{i,j,a,t}, \quad <7.23>$$

Finalmente, hay que destacar que el efecto del proceso de gating, típicamente incluido en el JPDA, se observa en el cálculo de los valores de verosimilitud $p(\bar{y}_t | \bar{x}_{j,t}^{(a)}) / a=1:n, j=1:k_{out,t}$ de la ecuación <7.20>, y está implícito en el de las variables $p_{i,j,a,t}$ y $d_{i,j,a,t}$, pues éstas se definen sólo para los $\bar{y}_{i,t} / i=1:m_{j,a,t}$ ubicados dentro del gate de cada partícula $\bar{x}_{j,t}^{(a)}$ relacionada con cada target j , tal y como éste se define en el apartado 7.2.1.

El proceso de gating comentado disminuye, por tanto, de forma importante, el coste computacional del proceso de asociación en el S-JPDAF. Además, este proceso se usa como base para incorporar y eliminar hipótesis de seguimiento en el S-JPDAF, mediante un algoritmo equivalente al planteado en el apartado 7.2.1 para el JPDAF:

1. Se crea una nueva hipótesis de estimación, $k_{out,t} = k_{out,t-1} + 1$ (se inicializa un nuevo PF) por cada medida $\bar{y}_{i,t} / i=1:m_t$ que se ubique fuera del gate de todas las n representaciones

de las $k_{out,t-1}$ hipótesis existentes. Para ello se genera un conjunto de partículas $S_{k_{out,t},t} = \left\{ \bar{x}_{k_{out,t},t}^{(a)}, 1/n \right\}_{a=1}^n$ asociado a cada nuevo objeto $k_{out,t}$ en función del valor de cada medida no asociada $\bar{x}_{j,t}^{(1:n)} = g(\bar{y}_{i,t}, \bar{v})$.¹¹

2. Se elimina la hipótesis de estimación j , $k_{out,t} = k_{out,t-1} - 1$ (se elimina el PF correspondiente) si ninguna de las m_i medidas se ubica dentro del gate de cualquiera de las n representaciones que la hipótesis j tiene en el PF correspondiente ($m_{a,j,t} = 0 / a = 1:n$).

El algoritmo S-JPDAF usado para realizar la comparativa perseguida en este apartado se completa, por tanto, con la etapa de creación y eliminación de hipótesis descrita, y su funcionalidad global se muestra en la Figura 7.9. En esta figura, al igual que en la Figura 7.1 que describe la funcionalidad del JPDAF, las etapas del núcleo de estimación incluido en el S-JPDAF (el PF) se muestran en color naranja, la de su núcleo de asociación JPDA en malva, y los pasos del proceso de modificación del número de hipótesis, en verde.

La similitud funcional de las dos propuestas basadas en el JPDA (la continua JPDAF, y muestreada S-JPDAF, diferenciadas únicamente en el núcleo de estimación) se hace evidente al comparar las Figuras Figura 7.1 y Figura 7.9. Esta semejanza funcional permite obtener, en el punto siguiente, conclusiones sobre el comportamiento empírico del S-JPDAF a partir las expuestas en el apartado 7.2.2 sobre el del JPDAF.

En [Schulz03b] se propone un método de creación y eliminación de hipótesis basado en la estimación de la PDF $p(k_{out,t} | G_{1:k,t|in})$ mediante un proceso MAP. Este método requiere un procesado previo del conjunto de medidas Y_t , semejante al de agrupación propuesto en esta tesis ($Y_t \Rightarrow G_{1:k,t|in}$), para obtener un set de características específicas $G_{1:k,t|in}$ con el que estimar de forma robusta el número de targets implicados en cada instante en la tarea de MTT.

El proceso descrito en el párrafo anterior comparte el planteamiento expuesto en esta tesis de que el uso de un proceso determinístico de clasificación de medidas ($Y_t \Rightarrow G_{1:k,t|in}$) robustece el comportamiento del algoritmo de MTT. Sin embargo, en la propuesta presentada en [Schulz03b] el conjunto de características $G_{1:k,t|in}$ extraído del set Y_t no se utiliza en el resto de etapas del lazo de ejecución de los PFs, como se propone en el XPFCP, por lo que el S-JPDAF resultante no incluye las ventajas (mostradas a lo largo de este capítulo), que aporta la inserción del proceso determinístico en el sistema de seguimiento multi-objeto.

¹¹ En el problema de MTT aquí expuesto, y teniendo en cuenta el modelo CV elegido para su desarrollo, el proceso $g(\bar{y}_{i,t})$ consiste en añadir al valor de la medida no asociada $\bar{y}_{i,t}$ dos componentes de valor nulo que representen inicialmente la velocidad de la hipótesis caracterizada por las partículas $[v_x \ v_z]^T = [0 \ 0]^T$. Además, para diversificar el set $S_{k_{out,t},t}$ el conjunto de partículas se distribuye $\bar{x}_{k_{out,t},t}^{(1:n)} = g(\bar{y}_{i,t}, \bar{v}_t)$ alrededor de ese valor a través del ruido de estado usado \bar{v}_t , que en el modelo CV es gaussiano, blanco y de media nula: $norm(0, V)$.

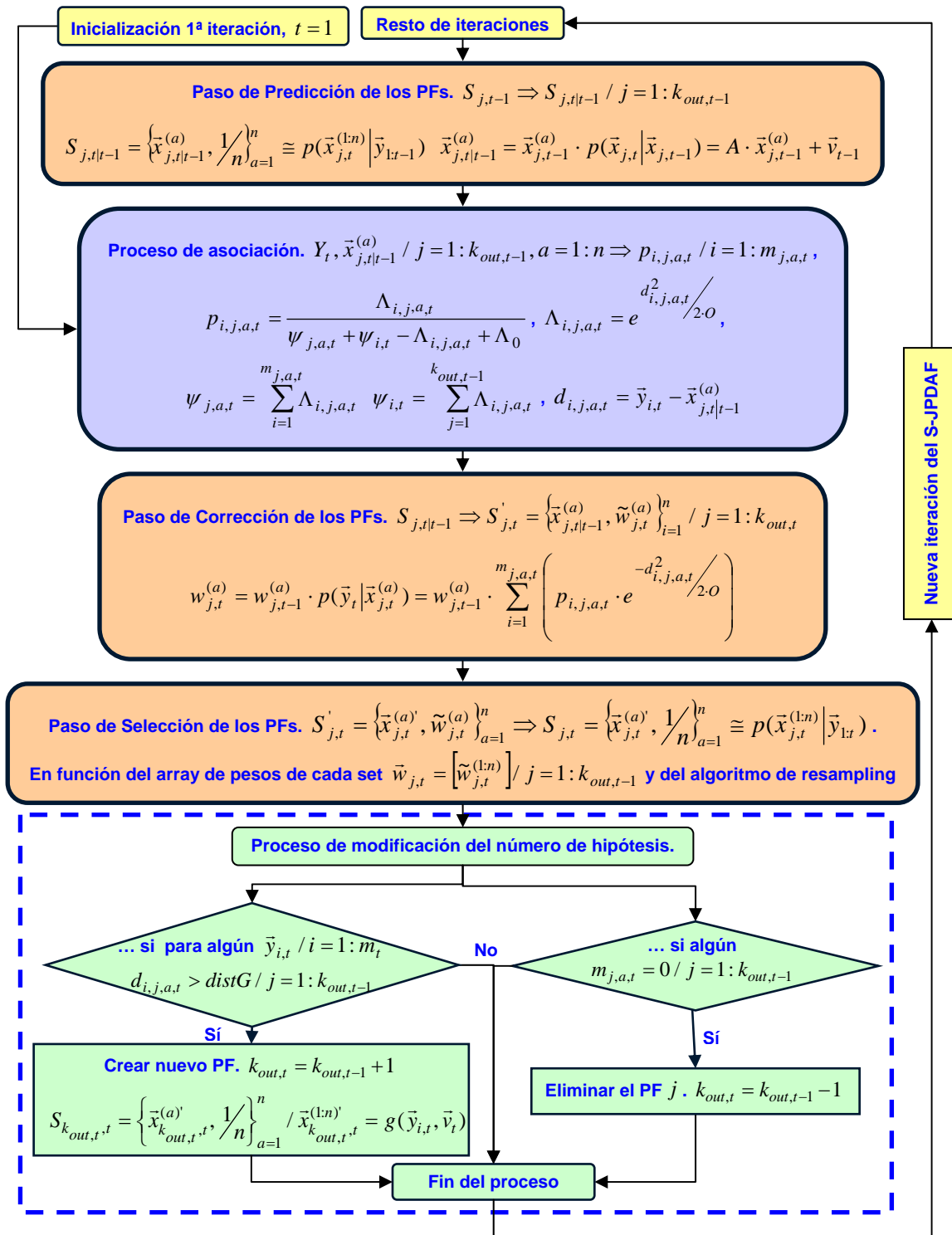


Figura 7.9. Diagrama funcional del algoritmo S-JPDF aplicado a la tarea de MTT de interés. Las etapas del núcleo de estimación (los PFs) se muestran en color naranja, la del algoritmo de asociación JPDA en malva, y los pasos del proceso de modificación del número de hipótesis, basado en el gating, en verde.

Una las ventajas referenciadas en el párrafo anterior, es el robustecimiento de la multimodalidad de la PDF discreta que representa el conjunto de partículas, gracias al cual el XPFCP genera como salida una estimación conjunta del estado de todos los $k_{out,t}$ objetos

implicados en la tarea de MTT con un solo PF $S_t = \left\{ \bar{x}_t^{(a)}, 1/n - n_{m,t} \right\}_{a=1}^{n-n_{m,t}} \cong p(\bar{x}_t^{(1:n)} | \bar{y}_{1:t})$. La salida del S-JPDAF es, sin embargo, un conjunto de PDFs unimodales discretas $S_{j,t} = \left\{ \bar{x}_{j,t}^{(a)}, 1/n \right\}_{a=1}^n \cong p(\bar{x}_{j,t}^{(1:n)} | \bar{y}_{1:t}) / j = 1:k_{out,t}$ (una por cada objeto), de las que se obtiene (como solución determinística del problema de MTT) el estado más probable de cada objeto mediante alguna medida probabilística realizada sobre el set correspondiente.

Del análisis funcional incluido en este apartado se concluye que el S-JPDAF, al igual que la versión no muestreada del algoritmo (el JPDAF), no alcanza las especificaciones planteadas al principio de la tesis para la resolución de la tarea de MTT de interés. El S-JPDAF incluye como núcleo de estimación (al igual que el XPFCP propuesto en la tesis) un PF que aporta al proceso de seguimiento multi-objeto la flexibilidad que el JPDAF no tiene.

El problema principal del S-JPDAF en su aplicación a la tarea referida es, sin embargo, la alta carga computacional del algoritmo de asociación JPDA incluido, que en la versión muestreada del JPDAF se multiplica por cada representación discreta de las distintas hipótesis de seguimiento (por cada partícula del PF relacionada con cada target). El proceso de clasificación propuesto en esta tesis simplifica este problema de asociación, como se demuestra en el capítulo 6, y permite utilizar una versión más flexible y robusta del filtro de Bayes que el KF, manteniendo la especificación de tiempo real también perseguida en la tesis.

En el apartado siguiente se demuestran los comentarios expuestos en el párrafo anterior y se completa de forma empírica la comparativa teórica entre el XPFCP y el S-JPDAF abordada en este punto. El análisis empírico expuesto a continuación destaca otras características del XPFCP, al margen de su bajo coste computacional, que lo hacen más adecuado en aplicaciones de MTT que el S-JPDAF típicamente usado ([Schulz03b], [Frank03] o [Almeida05]).

7.2.4. Resultados comparativos del S-JPDAF frente al XPFCP

En este punto, el análisis teórico sobre el S-JPDAF, incluido en el apartado anterior, se complementa con un conjunto de resultados obtenidos con este seguidor multi-objeto en las mismas pruebas realizadas en el capítulo 6 sobre el XPFCP, para completar así la comparativa funcional entre la propuesta de MTT planteada en la tesis y el S-JPDAF. Este análisis empírico es, por tanto, equivalente al incluido en el apartado 7.2.2. sobre la versión no muestreada del JPDAF.

La similitud funcional entre el S-JPDAF y el JPDAF, comentada en el punto anterior, lleva a no incluir en este apartado algunos de los resultados obtenidos con el S-JPDAF y referenciar, en su lugar, a los mostrados en el apartado 7.2.2.

a) Sensibilidad y ajuste de los parámetros del JPDAF

Las pruebas de sensibilidad del S-JPDAF respecto a sus parámetros de gating ($distG$) y ruido de medida (O que define el modelo de este ruido, idéntico al usado en el JPDAF) arrojan resultados similares a los expuestos en el mencionado apartado 7.2.2.

En lo referente al $distG$, el S-JPDAF presenta el mismo problema de alta sensibilidad al valor del gate que tiene el JPDAF, inexistente en el XPFCP, y que se analiza a continuación:

- Las pruebas realizadas muestran que no existe una zona de ajuste estable para $distG$, es decir, al aumentar el valor del parámetro aumenta considerablemente la tasa de errores por unión de 2 hipótesis en 1 (con una desviación típica de este tipo de error del 16% en una modificación del parámetro del 30% respecto al valor de ajuste), y disminuye de la misma forma la tasa de error por duplicación o desplazamiento de hipótesis (con una desviación típica de este tipo de error del 29% ante la misma modificación del valor de $distG$). Este comportamiento del S-JPDAF disminuye su robustez y flexibilidad para hacer frente a situaciones de tracking complejas (como le ocurre al JPDAF, según muestran los resultados expuestos en el punto 7.2.2), que el XPFCP resuelve, sin embargo, correctamente.
- Por otro lado, el valor de ajuste de $distG$ que permite alcanzar un comportamiento óptimo del S-JPDAF coincide con el obtenido para este parámetro en el JPDAF: $distG_{opt} = 640mm$.

En lo referente a la sensibilidad del S-JPDAF frente a modificaciones del valor de la matriz de covarianza del ruido de medida O ¹², este seguidor también muestra un comportamiento semejante al de su versión no muestreada (JPDAF): la sensibilidad del seguidor en este caso es también alta, como también lo es la equivalente del XPFCP, como se analiza a continuación:

- La Figura 7.10 muestra la tasa de error generada por el S-JPDAF en función del valor asignado a la desviación típica que define las componentes de la diagonal principal de la matriz O ($\sigma_{o,i} / i = \{x, y, z\}$), ante una variación de $\sigma_{o,i} = [75, 225]mm$ y bajo las mismas condiciones de test usadas para obtener las Figuras 6.21 del capítulo 6 (con el XPFCP) y Figura 7.3 (con el JPDAF).
- En esta Figura 7.10 se observa que la desviación típica de la tasa de error es de media entre 7 y 13 puntos superior a las del JPDAF y del XPFCP, respectivamente, y que el valor de ajuste óptimo de $\sigma_{o,i}$ coincide con el obtenido para él en el JPDAF: $\sigma_{o,i,opt} = 100mm$. La única diferencia que aparece en este análisis (Figura 7.10) frente al equivalente realizado sobre el JPDAF (Figura 7.3), es que los resultados generados por el S-JPDAF en los dos tests son semejantes, mientras que, en los arrojados por el JPDAF, la sensibilidad comentada es más evidente en situaciones complejas de MTT (test complejo) que en otras más normales (test completo corto). Este hecho, que también se observa en el análisis equivalente realizado al XPFCP (ver en la Figura 6.21 del capítulo 6), demuestra que la flexibilidad del núcleo de estimación del S-JPDAF (un PF como el usado en la propuesta de MTT planteada en esta tesis) es mayor que la del núcleo del JPDAF (un KF).

¹² El ajuste de los parámetros O , V y n del S-JPDAF que se expone a continuación se ha realizado en orden inverso al de explicación (en primer lugar se ajustó n , después V , y finalmente O), por lo que las tasas de error conseguidas con el ajuste óptimo de cada parámetro no disminuye, sino que aumentan a lo largo de la exposición. La tasa de error mínima conseguida, una vez ajustados todos los parámetros es correcta, y se corresponde con la obtenida, por tanto, con el ajuste de O .

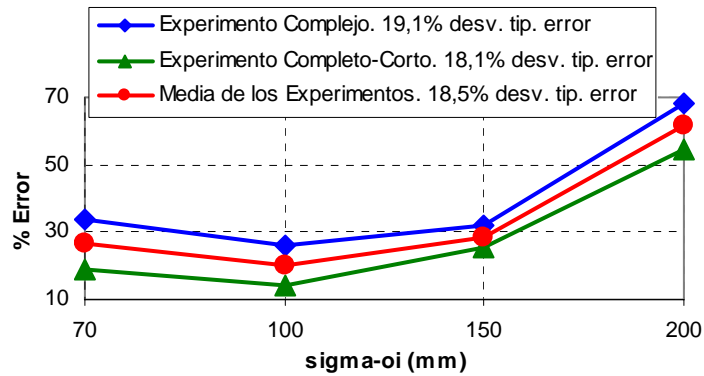


Figura 7.10. Tasa de error generada por el S-JPDAF en función de la desviación típica ($\sigma_{o,i}$, sigma-oi) de las componentes del vector de ruido de medida \vec{O}_i , supuestas gaussianas, incorreladas y de media nula, en el test complejo (en azul) y en el completo corto (en verde), y valor medio de ambos resultados (en rojo). Esta figura es equivalente a las Figuras 6.21 del capítulo 6 y Figura 7.3, que muestran la sensibilidad del XPFCP y del JPDAF, respectivamente, en estas condiciones.

El análisis de sensibilidad del S-JPDAF respecto a los valores de desviación típica $\sigma_{v,i} / i = \{x, y, z, vx, vz\}$ que definen, a través de la matriz V , el modelo de error de estado usado en este seguidor multi-objeto, se realiza al margen del estudio anterior, debido a que la ecuación de estado, y por ende, el modelo de ruido asociado a la misma, es diferente en las versiones continua y muestreada del JPDAF (en el JPDAF se usa el modelo \widehat{CV} y en el S-JPDAF, al igual que en el XPFCP, se emplea el CV, como se explica en el punto 7.2.1).

La Figura 7.11 permite analizar la sensibilidad del S-JPDAF frente a modificaciones del parámetro $\sigma_{v,i}$, mostrando la tasa de error generada por él en las mismas condiciones de test usadas para analizar esta sensibilidad en el XPFCP (Figura 6.19 del capítulo 6) y en el JPDAF (Figura 7.2). La conclusión que se extrae de la Figura 7.11 es igual a la obtenida del análisis equivalente realizado sobre los otros dos seguidores: el S-JPDAF no muestra una alta sensibilidad respecto al valor de este parámetro. Además, a la vista de los resultados mostrados en la Figura 7.11, el valor de ajuste óptimo de $\sigma_{v,i}$ coincide con el obtenido para el equivalente en el XPFCP y el JPDAF: $\sigma_{v,i|opt} = 100mm$.

Para finalizar el análisis de sensibilidad del S-JPDAF queda comprobar como cambia su comportamiento ante variaciones del número de partículas n usado para cada PF incluido en el seguidor (un PF por cada hipótesis de seguimiento, como se explica en el apartado 7.2.1).

La Figura 7.12 muestra la tasa de error generada por el S-JPDAF en función de n en los mismos experimentos usados para obtener las figuras Figura 7.10 y Figura 7.11. Esta Figura 7.12 es equivalente a la Figura 6.4 del capítulo 6 obtenida para analizar la sensibilidad del XPFCP frente a este mismo parámetro. En la Figura 7.12, sin embargo, solo se realiza el estudio hasta un valor de $n = 800$ pues el algoritmo se hace muy lento (como se expone en el apartado 7.2.1 y se demuestra en un punto posterior) y, tal y como se desprende de la figura, la fiabilidad del seguidor multi-objeto no mejora al aumentar n por encima de 600 partículas.

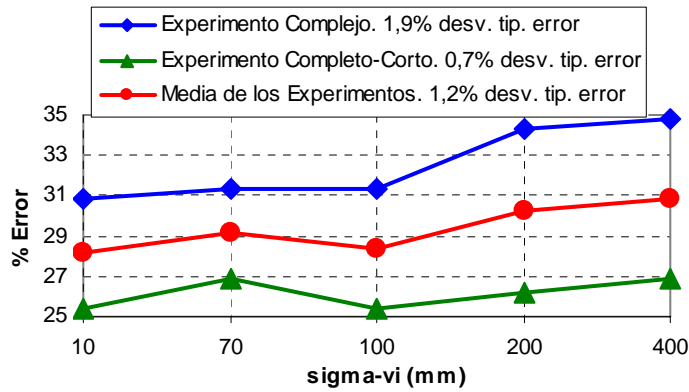


Figura 7.11. Tasa de error generada por el S-JPDAF en función de la desviación típica ($\sigma_{v,i}$, sigma-oi) de las componentes del ruido de estado \bar{v}_t en las condiciones de test de la Figura 7.10. Esta figura es equivalente a las 6.19 del capítulo 6 y Figura 7.2 al mostrar la sensibilidad del XPFCP y del JPDAF, respectivamente.

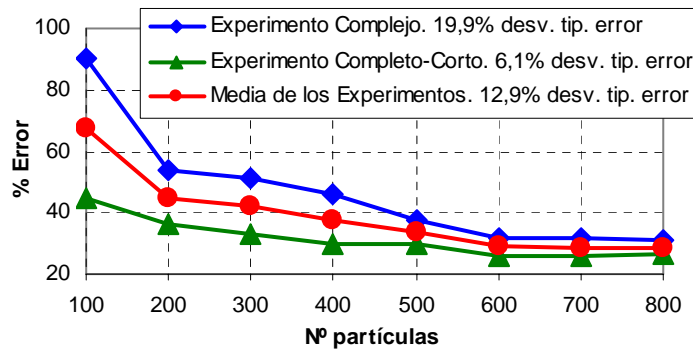


Figura 7.12. Tasa de error generada por el S-JPDAF en función del número de partículas n de cada PF incluido en el seguidor, al ejecutarlo en el experimento complejo (en azul) y en el completo corto (en verde), así como la media de ambos resultados (en rojo). Esta figura es equivalente a Figura 6.4 del capítulo 6 obtenida para el XPFCP.

De este análisis de sensibilidad se obtienen otras conclusiones, al margen de las expuestas en el párrafo anterior, que se listan a continuación:

- La conclusión principal que se extrae del análisis de la Figura 7.12 es que la fiabilidad del S-JPDAF es muy sensible al valor de n , mientras que la del XPFCP no lo es prácticamente: en la figura, la tasa de error obtenida con el S-JPDAF presenta una desviación típica media del 12.9%, frente a la del 1.3% dada por el XPFCP en las mismas condiciones (ver la Figura 6.4 del capítulo 6).
- Por otro lado, teniendo en cuenta además, que la tasa de error del S-JPDAF se mantiene constante, independientemente del número de partículas usado en cada PF, a partir de 600, se usa ese valor como ajuste óptimo del parámetro $n_{opt} = 600$. Es interesante resaltar el hecho de que ése es también el valor de ajuste óptimo usado para n en el XPFCP. De este hecho se deduce que el seguidor multi-objeto propuesto en esta tesis trabaja de forma óptima con el mismo número de partículas para todo el conjunto de objetos implicados en las pruebas realizadas, que el que requiere el S-JPDAF en cada uno de los PFs que incluye (tantos como objetos) para su funcionamiento óptimo.

- La Figura 7.13 muestra el t_{exe} del S-JPDAF en las mismas pruebas mostradas en la Figura 7.12, en función del valor de n , en un experimento equivalente al realizado sobre el XPFCP cuyos resultados se muestran en la Figura 6.6 del capítulo 6. Del análisis de la Figura 7.13 se extraen dos conclusiones principales: el t_{exe} del S-JPDAF crece de forma exponencial con el número de partículas de cada PF incluido en él; el seguidor no se ejecuta en tiempo real en el experimento complejo para un valor de $n \geq 200$, incumpliendo una de las especificaciones establecidas en la tesis. Este análisis temporal se completa en un punto posterior, sin embargo la figura referida se incluye en este punto para demostrar esta problemática

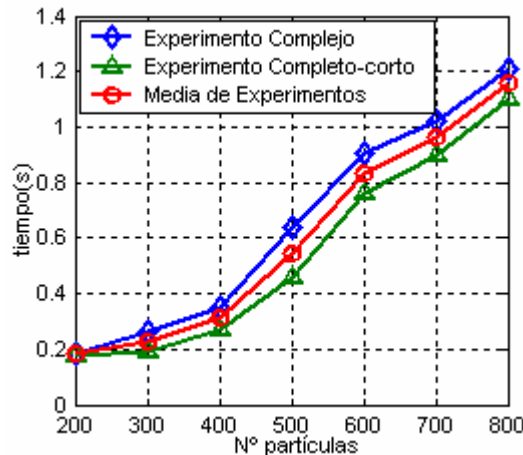


Figura 7.13. Gráfica comparativa del t_{exe} del S-JPDAF en función de n , en los mismos experimentos de la Figura 7.12. Esta figura es equivalente a la Figura 6.6 del capítulo 6, obtenida para el XPFCP.

El análisis expuesto confirma que el XPFCP es más robusto que el S-JPDAF pues, al contrario que este último, su funcionamiento es insensible al valor de ajuste del parámetro de gating ($disG$), y del número de partículas (n). Los resultados expuestos muestran que la alta sensibilidad del S-JPDAF respecto al valor de estos dos parámetros lleva a este seguidor a incumplir las especificaciones de fiabilidad y t_{exe} planteadas al principio de la tesis y alcanzadas, sin embargo, por el XPFCP.

b) Funcionalidad, fiabilidad y robustez frente al ruido

Realizado el ajuste de los parámetros del S-JPDAF se aborda la comparación de este algoritmo frente a la del XPFCP, desde el punto de vista de la fiabilidad.

Como base de esta comparación se obtiene Figura 7.14 que muestra la tasa de error generada por ambos algoritmos (en azul para el XPFCP y en verde para el S-JPDAF) en los experimentos complejo (a), completo corto (b) y de ruido, en función del tipo de error, de forma equivalente a lo mostrado en la Figura 7.5, para comparar el XPFCP y el JPDAF desde este punto de vista.

Las conclusiones extraídas de la Figura 7.14 son muy semejantes a las obtenidas de la referenciada Figura 7.5. A continuación se exponen las más importantes:

- El S-JPDAF presenta, al igual que el JPDAF (ver la Figura 7.5) una fiabilidad algo menor a la que muestra el XPFCP (genera una tasa de error media 3.5 puntos mayor). Esta diferencia

es sobre todo evidente en situaciones difíciles de tracking como las presentes el test complejo, en las que el S-JPDAF genera más errores de todos los tipos que el XPFCP. En situaciones de complejidad diversa (como las que se dan en el test completo corto) la diferencia no es tan importante, ya que se debe a errores de duplicación de hipótesis, menos relevantes en la aplicación de MTT que los de unión de 2 hipótesis en 1.

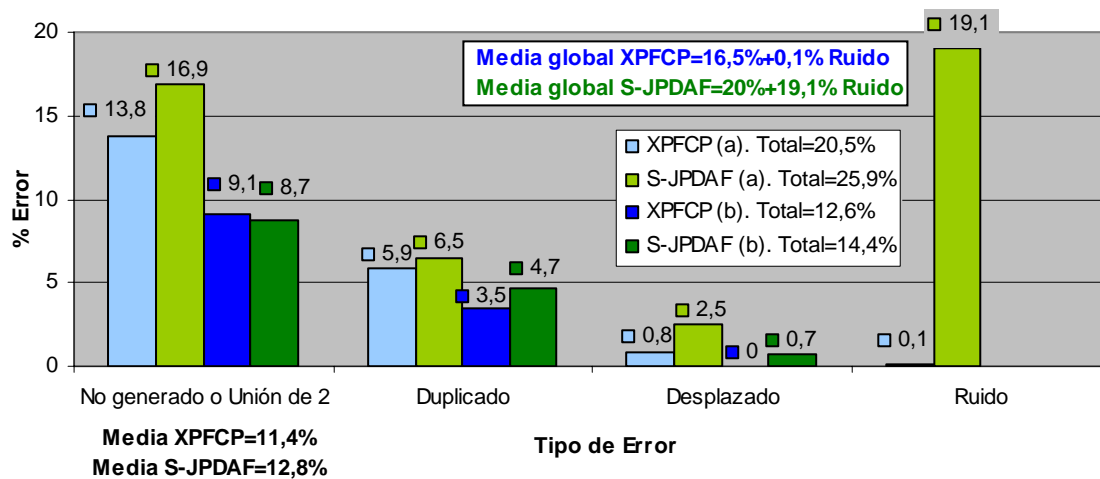


Figura 7.14. Tasas de error generadas por el JPDAF (en verde) y el XPFCP (en azul) en función del tipo de error, en los tests complejo (tonos claros), completo corto (tonos oscuros) y de ruido (para la tasa de ruido no filtrado).

- La tasa de ruido no filtrado generada por el S-JPDAF en el experimento de ruido es, al igual que la generada por el JPDAF, mucho mayor que la generada por el XPFCP. Ninguno de los seguidores basados en el JPDA incluye un mecanismo de rechazo al ruido, como el que sí incluye el XPFCP a través de la validación de clases de medidas. Debido a este hecho, el JPDAF en sus versiones continua y discreta solo filtra los outliers que permanecen durante una sola iteración de proceso de seguimiento, por la razón ya expuesta en el apartado 7.2.2.

La conclusión principal que se extrae del análisis de la Figura 7.14 es que el XPFCP, gracias a las mejoras introducidas a través del proceso de clasificación de medidas y en el propio lazo de ejecución del PF, aporta una mayor fiabilidad y robustez al ruido que el S-JPDAF, que se evidencia sobre todo, en situaciones complejas de tracking.

No se incluye en este apartado un análisis de fiabilidad del S-JPDAF en la identificación de las clases, como el expuesto en la comparativa empírica entre el XPFCP y el JPDAF, pues la versión muestreada del JPDAF, al igual que la continua, no incluye ningún proceso de identificación específico, por lo que muestra un comportamiento muy semejante al del JPDAF, mostrado en la Figura 7.6 (confunde la identidad de los objetos en los cruces). El comportamiento del XPFCP propuesto mejora el del S-JPDAF también en este aspecto.

Con objeto de completar la comparación de fiabilidad entre el XPFCP y el S-JPDAF, en la Figura 7.15 se muestra el valor de factor de calidad \hat{n}_{eff} (empleado en el capítulo 6 para asegurar la convergencia del set S_t) que arrojan los dos algoritmos (en verde para el S-JPDAF y en azul para el XPFCP) en el test complejo. En la figura se muestra el valor instantáneo de \hat{n}_{eff} en línea fina, y en trazo grueso el valor medio, de los diferentes PFs en el caso del

S-JPDAF. Además, a la derecha de la gráfica se añade una tabla que indica el valor numérico medio de este factor de calidad generado por los dos seguidores en los experimentos complejo (a) y completo corto (b). Las conclusiones extraídas de esta figura se listan a continuación:

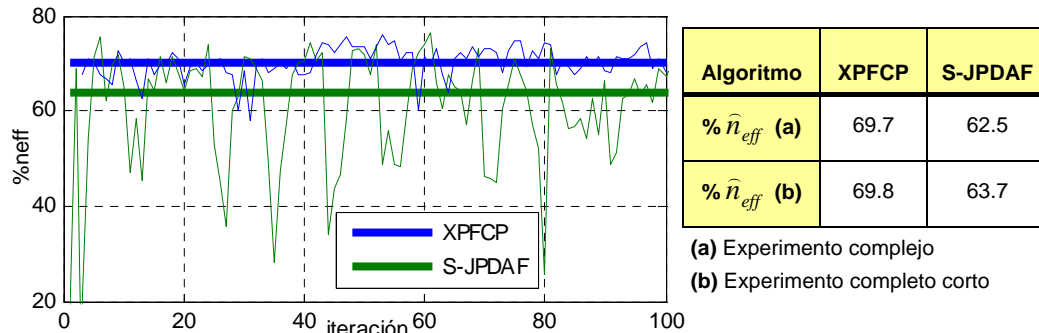


Figura 7.15. Gráficas comparativas del valor de \hat{n}_{eff} generado en el experimento complejo por el S-JPDAF y el XPFCP. Las líneas gruesas muestran el valor medio de las de trazo fino. Se incluye una tabla con el valor medio de \hat{n}_{eff} en ese test (a) y en el completo corto (b).

La Figura 7.15 muestra que ambos algoritmos mantienen un valor de \hat{n}_{eff} suficientemente alto, y similar en ambos tests, como para asegurar que el/los sets de partículas no divergen¹³.

Sin embargo, como se observa en la Figura 7.15, el XPFCP arroja un valor medio algo mayor de este factor de calidad que el S-JPDAF, debido a que no muestra tanta variación en él a lo largo del tiempo (picos decrecientes en la gráfica verde de trazo fino). Las continuas modificaciones del valor de \hat{n}_{eff} generado por el S-JPDAF se producen coincidiendo con la creación o eliminación de algún PF, que, como se observa en la gráfica referida, tienen inicialmente un valor bajo de este factor de calidad, por lo que la media global de \hat{n}_{eff} generada por el S-JPDAF disminuye. El factor \hat{n}_{eff} generado por el XPFCP mantiene un valor medio mayor al del S-JPDAF, por tanto, debido a que la creación y eliminación de hipótesis se realiza en un solo PF, gracias al paso de re-inicialización (ver en el apartado 6.1.1 del capítulo 6).

El XPFCP muestra también, por tanto, un mejor comportamiento que el S-JPDAF desde el punto de vista del factor de calidad \hat{n}_{eff} .

c) Tiempo de ejecución

La última comparación empírica realizada en este apartado entre el XPFCP y el S-JPDAF se refiere a la carga computacional de los dos algoritmos.

La Figura 7.16 muestra el t_{exe} de ambos seguidores en el experimento complejo (el del XPFCP en azul y el del JPDAF en verde oscuro), y el del JPDAF, en el mismo experimento, extraído de la Figura 7.7 (en verde claro). La Figura 7.16 muestra que el t_{exe} del S-JPDAF es

¹³ Se suele considerar que el set S_t no sufre empobrecimiento si $\hat{n}_{eff} > \frac{2}{3} \cdot n$ ([Gustaffson02], capítulo 4).

mucho mayor que el de los otros dos algoritmos (de media, 19 veces el del XPFCP y casi 5 el del JPDAF). Este hecho se debe a que el coste computacional del S-JPDAF es incluso mayor al del JPDAF, pues en la versión muestreada del JPDAF cada hipótesis de seguimiento tiene n representaciones (una por cada partícula del PF al que se asocia) en lugar la única (caracterizada por el KF al que se asocia) que tienen en la versión continua del mismo algoritmo.

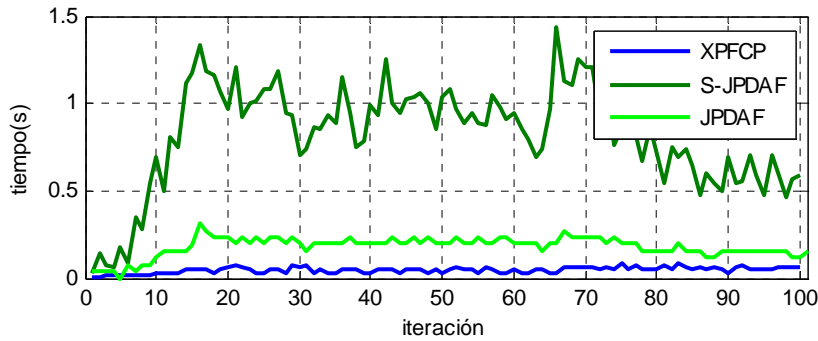


Figura 7.16. Gráficas comparativas del t_{exe} del S-JPDAF (en verde oscuro), del XPFCP (en azul) y del JPDAF (en verde claro) en el experimento complejo.

Al margen de la carga computacional de los $k_{out,t}$ PFs incluidos en el S-JPDAF, el proceso de asociación de tipo JPDA consume la mayor parte del t_{exe} del S-JPDAF. De hecho, el JPDA en el S-JPDAF pasa a tener una complejidad computacional n veces mayor a la que tiene en la versión continua del algoritmo ($O((k_{out,t} + 1)^{m_t})$), y por ende mucho mayor a la del proceso de asociación del XPFCP $O(n^{k_{in,t}}) / k_{in,t} \ll m_t$: $O((n \times k_{out,t} + 1)^{m_t})$. La Tabla 7.3 muestra como se reparte de forma absoluta y relativa, el t_{exe} del cada uno de los tres seguidores multi-objeto, en el experimento complejo, entre los procesos de estimación y asociación incluidos. Esta tabla es equivalente a la Tabla 7.1, y confirma que la mayor parte del tiempo de cómputo del S-JPDAF lo consume el proceso de asociación JPDA, y que este tiempo es relativamente mayor que el que consume el mismo proceso en el JPDAF.

Tabla 7.3. Valor medio absoluto y relativo del tiempo de cómputo de los procesos de asociación y estimación implicados en el S-JPDAF, el JPDAF y el XPFCP al ser ejecutados en el experimento complejo.

Clasificador de medidas	t_{exe} total (s)	t_{exe} asociación (s, % del total)	t_{exe} de estimación (s, % del total)
XPFCP	0.045	0.013, 29%	0.032, 71% (36% del PF + 35% del cluster de partículas)
S-JPDAF	0.859	0.779, 91%	0.080, 9%
JPDAF	0.179	0.143, 79%	0.036, 21%

Por otro lado, la Figura 7.17 muestra el t_{exe} de los tres algoritmos analizados (en verde oscuro el del S-JPDAF, en verde claro el del JPDAF y en azul el del XPFCP) en función del número de objetos seguidos en el experimento complejo largo. Esta figura es equivalente a la Figura 7.8 y confirma que sólo en el caso del XPFCP el t_{exe} del seguidor se mantiene

prácticamente constante con $k_{out,t}$. El t_{exe} de los dos algoritmos basados en el JPDA crece, sin embargo, exponencialmente (pero con distinta ganancia) con $k_{out,t}$, por lo que el algoritmo no se puede ejecutar en tiempo real, como se especifica en la tesis, en aplicaciones de MTT con más de tres o cuatro objetos.

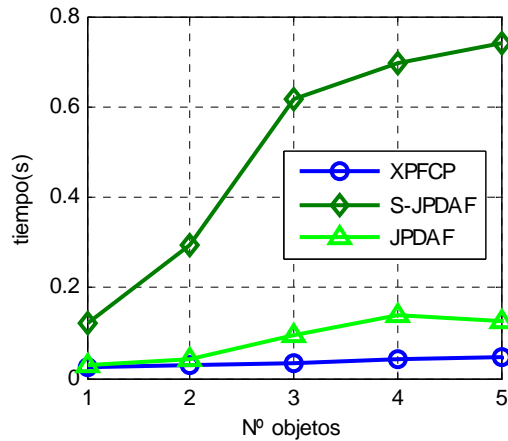


Figura 7.17. Gráficas comparativas del t_{exe} del S-JPDAF (en verde oscuro), del JPDAF (en verde claro) y del XPFCP (en azul) en el experimento completo largo.

La Tabla 7.4 confirma, finalmente, lo expuesto en los párrafos anteriores. En ella se muestra la velocidad media de ejecución de los tres algoritmos referidos, en las mismas condiciones de la Tabla 7.2, a partir de la que se concluye que el XPFCP es el único algoritmo de entre estos tres que se ejecuta en tiempo real en cualquier tipo de situación (su velocidad de ejecución es siempre mayor a 15FPS).

Tabla 7.4. Velocidad media de ejecución del S-JPDAF, del JPDAF y del XPFCP en los experimentos complejo (a), completo corto (b) y completo largo (c)

Algoritmo/Test	FPS(a)	FPS(b)	FPS(c)
XPFCP	22	23.6	25
S-JPDAF	1.2	1.9	3.2
JPDAF	5.6	8	11

Los análisis de sensibilidad, robustez, fiabilidad y t_{exe} expuestos en este punto, permiten concluir globalmente que el XPFCP presenta un mejor comportamiento que el S-JPDAF en la tarea de MTT de interés, y que al contrario que las versiones continua y muestreada del JPDAF, cumple todas las especificaciones planteadas en el capítulo 1.

7.3. Conclusiones sobre las comparativas expuestas

En este último apartado se resumen las conclusiones expuestas en los anteriores sobre la comparación teórica y empírica entre la propuesta de seguimiento de múltiples objetos

planteada en la tesis (XPFCP) y las otras dos propuestas elegidas para la comparación por su éxito entre la comunidad científica del área (el JPDAF en sus versiones continua y discreta).

1. La fiabilidad de los tres algoritmos es bastante similar en situaciones normales de tracking (la tasa de error generada por el XPFCP y el JPDAF en el test completo corto es casi igual y solo 2 puntos superior en los resultados obtenidos del S-JPDAF). El XPFCP muestra, sin embargo, una mayor fiabilidad ante problemas complejos (la tasa de error generada por el XPFCP es en ese caso 5.5 y 7 puntos menor que la obtenida con el S-JPDAF y JPDAF, respectivamente). Este mejor comportamiento del XPFCP se debe al refuerzo de las hipótesis más débilmente sensadas que realiza en el algoritmo el paso de re-inicialización, a través de la clasificación de medidas $Y_t \Rightarrow G_{1:k,t|in}$ y del proceso de validación incluido en ella (ver apartado 6.1 del capítulo 6). Este último proceso de validación permite, además al XPFCP, mantener identificados a los objetos seguidos, incluso en situaciones complejas de occlusión (especificación que no cumplen los otros dos algoritmos, como demuestra la Figura 7.6) y alcanzar tasas de rechazo al ruido de casi el 100%, muy superiores a las del 84% y 81% obtenidas con el JPDAF y el S-JPDAF, respectivamente.
2. El XPFCP muestra mayor robustez a la modificación de sus parámetros que el S-JPDAF o el JPDAF, una vez ajustados éstos a su valor de funcionamiento óptimo. Este hecho es sobre todo evidente con el parámetro de gating de los tres algoritmos: $distG$ para el S-JPDAF y el JPDAF y $distM$ para el XPFCP. Por otro lado, el S-JPDAF muestra también una alta sensibilidad al número de partículas n , que no presenta el XPFCP (ver en el apartado 6.1.3 del capítulo 6), y que impide su ejecución en tiempo real.
3. Se comprueba que el t_{exe} de los dos algoritmos basados en el JPDA es mucho mayor que el del XPFCP (este último se ejecuta de media a 25FPS mientras que el JPDAF lo hace a 11FPS y el S-JPDAF a 3FPS). Además este tiempo permanece prácticamente constante en el XPFCP al aumentar el número de objetos implicados en el tracking $k_{out,t}$, mientras que en los algoritmos basados en el proceso de asociación JPDA crece exponencialmente con él. Sólo el XPFCP cumple, por tanto como aquí se demuestra, las especificaciones de tiempo real establecidas en la tesis.
4. En todos los experimentos, el XPFCP muestra un valor medio del factor de calidad \hat{n}_{eff} semejante al arrojado por el S-JPDAF (de casi el 70% en el XPFCP, y de entre el 62% y el 64% en el S-JPDAF), cumpliendo ambos el criterio normalmente establecido ([Kreucher05]) para asegurar la no degeneración del set de partículas.

Esta última conclusión es quizás la más importante, pues demuestra que la propuesta de MTT realizada en esta tesis alcanza el objetivo planteado de usar la multimodalidad del PF para realizar el seguimiento de un número variable de objetos con un solo filtro, con una eficacia igual o incluso superior a la de propuestas más normalmente empleadas para esta aplicación (basadas en el uso de un estimador por objeto y un algoritmo de asociación más robusto como el JPDA usado en el S-JDPAF o el JPDAF, más robusto que el NN usado en el XPFCP – [Rasmussen01]–), y con una eficiencia mucho mayor a la de éstas debido al valor bajo y constante de su t_{exe} . El XPFCP propuesto y diseñado constituye así una solución al problema de

divergencia que muestra el PF usado como estimador multimodal, que lleva a otros autores a proponer soluciones computacionalmente más complejas.

El estudio comparativo expuesto queda así completo. Como colofón, en la Figura 7.18 se incluye una secuencia de imágenes (ordenadas en el tiempo de izquierda a derecha) extraídas de las pruebas realizadas en tiempo real sobre los tres algoritmos (los generados por el JPDAF en la fila superior, los del S-JPDAF en la central, y los del XPFCP en la inferior).

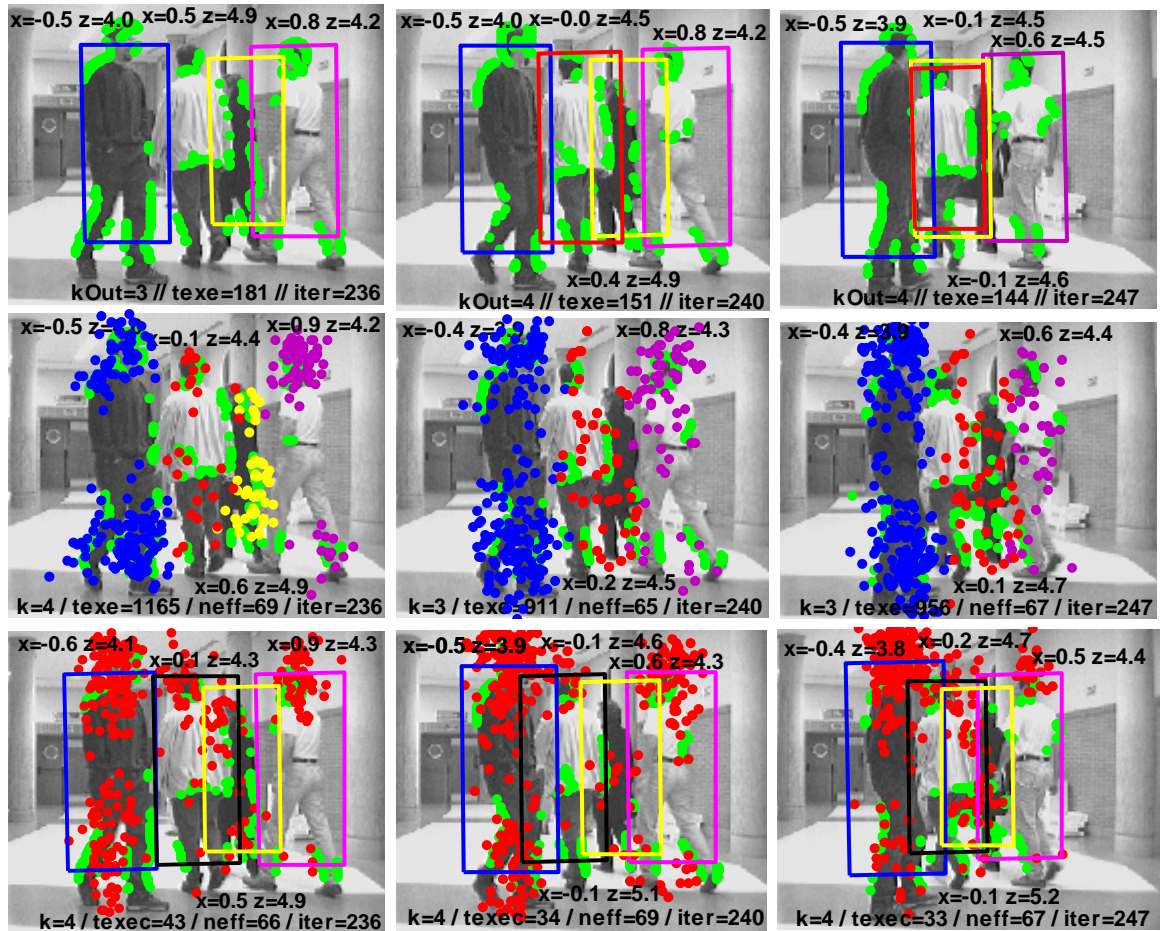


Figura 7.18. Proyección en la imagen de entrada de los resultados generados en el test complejo, por el JPDAF (fila superior de imágenes), el S-JPDAF (fila central) y el XPFCP (fila inferior). Se incluye el valor de la posición media estimada por cada seguidor para cada objeto detectado en las imágenes, y otros parámetros del proceso.

En esta Figura 7.18, al pie de la imagen de entrada al proceso de tracking en cada instante de test, se muestra la siguiente información sobre el proceso: el número de objetos seguidos $k_{out,t}$ (k o kOut); el t_{exe} del algoritmo correspondiente; el factor de calidad \hat{n}_{eff} obtenido (en el caso de los trackers basados en el PF); y el número de frame del experimento. Además, sobre cada imagen se muestra también la proyección de la solución de seguimiento generada para cada objeto por el algoritmo correspondiente, de la siguiente forma:

- En el caso del XPFCP (fila inferior de imágenes) esta solución se representa por colores en función de la identidad del objeto seguido $\tau_{j,t|out} / j=1:k_{out,t}$, mediante un cilindro orientado que proyectado en el plano imagen aparece como un rectángulo (ver el apartado 6.2.1 del capítulo 6. No se incluye la flecha indicadora de la velocidad de cada objeto para clarificar las imágenes. El valor del centroide de cada cilindro $\bar{g}_{j,t|out}$ se muestra sobre o bajo (para el objeto identificado en amarillo) el rectángulo correspondiente. La posición representada por cada partícula de S_t se muestra también en las imágenes con puntos rojos.
- En el caso del JPDAF (fila superior de imágenes) la salida del seguidor queda especificada por la gaussiana $norm(\bar{x}_{j,t}, P_{j,t})$ generada como solución de estimación por el KF asociado a cada objeto $j=1:k_{out,t}$. Siguiendo el mismo criterio planteado para el XPFCP en el apartado 6.2.1 del capítulo 6, esta solución puede representarse en el espacio de seguimiento XYZ mediante un cilindro (sin orientación, pues el estado del modelo \widehat{CV} usado en el JPDAF no incluye componentes de velocidad) de centroide $\bar{x}_{j,t}$ con varianza $P_{j,t}$, y radio $distG$, cuya proyección en el plano imagen es un rectángulo (coloreado en función de la identidad del KF asociado a cada objeto). En las imágenes correspondientes se muestra también el valor del centroide de cada cilindro.
- En el caso del S-JPDAF (fila central de imágenes) la salida del seguidor queda definida por el conjunto de partículas asociado a cada objeto $S_{j,t} / j=1:k_{out,t}$ (coloreadas en función de la identidad del PF asociado a cada objeto). La proyección en el plano imagen del valor de posición representado por cada una de ellas $\bar{x}_{j,t}^{(a)} / a=1:n$ se caracteriza por un punto. En este caso la posición estimada para cada objeto también se indica en cada imagen como el valor medio de posición representado por el conjunto de partículas asociado.

Las imágenes mostradas en la Figura 7.18 confirman las conclusiones expuestas en este punto: el t_{exe} del JPDAF en su versión continua y discreta es mucho mayor al del XPFCP; y en situaciones complejas (cruces entre objetos), estos algoritmos resuelven correctamente el problema de asociación pero no el de identificación de los objetos, debido a su alta sensibilidad al valor del parámetro de gating $distG$ y al hecho de que no incorporan un proceso de identificación como el que sí incluye el XPFCP implícitamente a través del proceso de validación del clasificador de partículas (ver apartado 6.2 del capítulo 6).

En la Figura 7.18 se observa un ejemplo de esta última afirmación: el JPDAF muestra un error de unión de 2 objetos en 1 sola hipótesis (la marcada en amarillo) en el frame 236 y otro de desplazamiento (del objeto marcado en amarillo, casi totalmente oculto por el marcado en rojo) en el frame 247; y el S-JPDAF genera errores del primero de los tipos en los frames 240 y 247 en la situación y por las razones comentadas. El XPFCP genera, sin embargo el resultado correcto en todos los casos¹⁴.

¹⁴ El objeto identificado en color rojo en los resultados generados por el JPDAF y el S-JPDAF se identifica en color negro en los del XPFCP, para no usar el mismo color con el que se representan las partículas en este último caso.

CAPÍTULO 8

CONCLUSIONES Y TRABAJOS FUTUROS A LA TESIS

En este último capítulo de la tesis se resumen las conclusiones extraídas del trabajo presentado en capítulos anteriores, así como las aportaciones principales realizadas con el desarrollo de la tesis.

Además, se incluye como final del capítulo un listado de líneas de investigación que abre el trabajo llevado a cabo en esta tesis.

8.1. Conclusiones extraídas del trabajo global expuesto

La primera conclusión extraída del desarrollo de la tesis, es la omnipresencia de trabajos y propuestas realizadas por la comunidad científica en el marco en el que ésta se encuadra: el seguimiento o tracking de un número variable de objetos $k_{out,t}$ en situaciones complejas. Esta tarea, como se explica en el capítulo 1, está incluida en gran cantidad de aplicaciones, muchas de ellas del área de robótica personal en el que se centra la tesis (vigilancia, detección de movimiento, HMIs, navegación de robots, etc.). La importancia de las aportaciones que se hagan en este tema, incluyendo las realizadas en esta tesis, queda así totalmente confirmada.

Dentro de las propuestas planteadas por la comunidad científica para resolver la tarea de MTT, lo habitual es que incluyan un algoritmo de estimación¹, y generalmente uno entre los dos siguientes: el KF o el PF. En esta tesis se concluye que, en términos de fiabilidad, ambos algoritmos generan resultados semejantes. Sin embargo, la propuesta planteada en esta tesis, basada en un PF, aporta ciertas ventajas frente a las basadas en el KF, en su aplicación a la tarea de MTT en situaciones complejas como las aquí analizadas: tiene un tiempo de ejecución prácticamente constante, independientemente del número de objetos $k_{out,t}$ a seguir; al generar un conjunto de estimaciones del problema en lugar de la óptima, generada por el KF, es menos inmune a los problemas de divergencia que puede presentar el KF ante ruidos y errores de asociación o de modelado.

Estas conclusiones son las razones principales que llevan a elegir el PF como algoritmo de estimación en la tesis, sin embargo, también hay que tener en cuenta la capacidad de este algoritmo de manejar modelos no lineales (no utilizados en la tesis) sin necesidad de incluir aproximaciones de éstos (a través del uso de versiones del KF como el EKF o el UKF) que disminuyan la fiabilidad y la velocidad de ejecución del seguidor. Este hecho, usado en muchos trabajos del área de MTT como única razón para la elección del PF, permite plantear, además, trabajos futuros basados en el algoritmo XPFCP propuesto (ver al final de este capítulo).

Otra conclusión extraída del desarrollo de este trabajo es la importancia del proceso de asociación (entre medidas de posición e hipótesis de seguimiento) del sistema de MTT en la correcta ejecución de la tarea de tracking. Debido a ello, algunas de las propuestas planteadas en el área de interés incluyen algoritmos de asociación complejos como el JPDA ([Hue02], [Schulz03b] o [Ng06]), mientras que otras se basan en técnicas más simples como el NN, y compensan la baja fiabilidad de éstas con la incorporación de modelos muy específicos de observación ([Khan05], [Branson05], [Kobayashi06], [Chang06]). En esta tesis se propone emplear un algoritmo sencillo de asociación (de tipo NN) manteniendo la flexibilidad del modelo de observación. Los resultados expuestos en los capítulos 6 y 7, demuestran que el algoritmo así definido alcanza la fiabilidad de propuestas más específicas y complejas.

Como conclusión final destacar que los resultados obtenidos con el algoritmo de MTT propuesto en esta tesis demuestran la posibilidad de usar la multimodalidad del PF para realizar de forma eficiente el seguimiento de un número variable de objetos $k_{out,t}$ en situaciones complejas. Esta posibilidad ha sido rechazada por la comunidad científica ([Hue02] o [Schulz03b]) en base a que trabajos como [Gordon97] o [Koller-Meier01], propuestos en esta línea, presentaban baja fiabilidad y robustez. Debido a ello se proponen algoritmos basados en el uso de un estimador por objeto ([Schulz03b], [Okuma04], o [Vermaak05a]) o un solo algoritmo que estime el estado del conjunto de todos los objetos a través de un modelo aumentado ([Isard01], [Spengler03] o [Smith05]). Las aportaciones realizadas en la tesis han demostrado, a través de los resultados expuestos, proporcionar al algoritmo de tracking planteado, la robustez y fiabilidad necesarias para resolver la tarea de MTT de interés con un sólo PF aprovechando su multimodalidad.

¹ Algunos trabajos llamados de tracking, sobre todo basados en visión, no incluyen algoritmo de estimación ([Yang03], [Khan06], [Fleuret08]), pero en ese tipo de propuestas no se consideran de seguimiento en esta tesis sino de segmentación, y por ello, no han sido analizadas en este documento.

8.2. Recapitulación de las aportaciones realizadas

A la vista de los análisis, desarrollos y resultados expuestos en la tesis, se puede resumir las aportaciones realizadas en esta tesis en el listado que se incluye a continuación:

1. Se ha diseñado un algoritmo para el seguimiento de un número variable de objetos $k_{out,t}$ a partir de la combinación de métodos probabilísticos y determinísticos. Concretamente, el XPFCP propuesto combina la extensión del PF planteada en [Koller-Meier01] con dos procesos de clasificación determinísticos para alcanzar el objetivo de estimar la posición y la velocidad de los $k_{out,t}$ objetos.
2. El estimador multimodal alcanza la robustez y fiabilidad necesaria para poder aplicarlo al problema de interés incorporando la información generada por un clasificador de medidas $Y_t \Rightarrow G_{1:k,t|in}$ diseñado al efecto (“*k-medias secuencial*” con validación, descrito en el capítulo 5), en el paso de corrección del PF y en uno nuevo de re-inicialización añadido al principio del lazo de ejecución del PF. Este proceso de clasificación facilita la incorporación y eliminación robusta de hipótesis de tracking en la creencia discreta $p(\bar{x}_{0:t}^{(i)} | \bar{y}_{1:t}) / i = 1:n$ modelada por el set S_t , de forma automática (sin necesidad de incluir un proceso específico de creación o eliminación de hipótesis, como los usados en [Smith05] o [Ng06]).
3. El proceso de clasificación booleano propuesto realiza, además, la tarea de asociación de forma intrínseca al proceso de estimación, con una fiabilidad equivalente y una eficiencia mayor a la de los algoritmos típicamente propuestos por la comunidad científica (p. ej. el MHT o el JPDA) para resolver el problema de asociación en aplicaciones de MTT como la de interés, como se demuestra en el capítulo 7.
4. El clasificador de medidas $Y_t \Rightarrow G_{1:k,t|in}$ robustece, además el comportamiento del seguidor, filtrando los outliers incluidos en el set de datos Y_t e insensibilizando de forma importante la fiabilidad del XPFCP del valor de sus parámetros.
5. Las mejoras propuestas en los pasos de re-inicialización y corrección del XPFCP tienen una funcionalidad de robustecimiento del set S_t ante el problema de empobrecimiento característico del PF (ver en el capítulo 4) semejante a la de otros algoritmos propuestos por la comunidad científica específicamente para ello (como el paso MCMC –[Liu98]–, el de “*Edición previa*” –[Gordon93]–, el de “*Rechazo Selectivo*” –[Pitt99]–, o el “*Muestreo Segmentado*” –[MacCormick00]–, descritos en el capítulo 4), pero con un coste computacional menor, y la funcionalidad, no incluida en estas otras propuestas, de adaptar la multimodalidad del PF a la estimación de múltiples sistemas con un solo filtro.
6. El XPFCP propuesto incluye la flexibilidad especificada en la tesis, que le permite ser aplicado a problemas de tracking basados en distintos tipos de sensores e incluso en la fusión de información procedente de varios de ellos (trabajo que queda abierto con el desarrollo de esta tesis). Los resultados mostrados en el capítulo 6, y los expuestos en [Marron04a] y [Marron04b] (se exponen resultados generados con el XPFCP en base a un sistema de sensores ultrasonidos) demuestran esta afirmación.

7. La flexibilidad comentada permite usar el algoritmo propuesto en el tracking de objetos de distinto tipo (rígidos o deformables, estáticos o dinámicos, etc.). El XPFCP genera para cada uno de ellos: la posición y velocidad de su centroide en el plano de movimiento XZ $\bar{g}_{j,t|out} = \{x_{j,t|out} \quad y_{j,t|out} \quad z_{j,t|out} \quad vx_{j,t|out} \quad vz_{j,t|out}\} / j = 1:k_{out,t}$; un identificador $\tau_{j,t}$ que, en situaciones complejas de seguimiento (cruces y oclusiones parciales), mantiene la identidad del objeto mientras éste permanece en el entorno de sensado; y un valor de certidumbre $p_{j,t}$ de la estimación que puede ser usado en un sistema superior de proceso (p.ej. un navegador) para validar los resultados de tracking generados.
8. El set S_t generado a la salida del XPFCP como representación probabilística de la creencia multimodal de seguimiento $p(\bar{x}_{0:t} | \bar{y}_{1:t})$ se clasifica en grupos para proporcionar una salida determinística al proceso de MTT. De los dos algoritmos diseñados para realizar esta tarea (uno booleano y otro borroso), el “*k-medias secuencial*” con validación presenta también un mejor comportamiento. Este proceso extrae del set a la salida del XPFCP la información referente a los objetos indicada en el punto anterior, con una fiabilidad del 98%. Esta alta fiabilidad se consigue, en gran medida, gracias al proceso de validación incluido en los dos clasificadores, que permite modular la velocidad de incorporación y eliminación de las hipótesis en el seguimiento y en su salida determinística (funciona de modo similar a las ventanas de validación temporal [Blackmann99]– usadas en soluciones de MTT como la descrita en [Ng06])
9. Por último, destacar que el XPFCP diseñado en la tesis tiene un tiempo de ejecución prácticamente constante e independiente del número de targets $k_{out,t}$ implicados en el tracking. Esta característica facilita su ejecución en tiempo real en aplicaciones complejas, en las que fallan otras propuestas analizadas en el capítulo 2.

8.3. Líneas de aplicación e investigación futuras

Finalizada esta tesis, es interesante destacar algunas de las líneas de trabajo o aplicación futura, así como las de investigación más importantes que se abren a partir de los resultados y algoritmos aquí propuestos.

Una vez demostrada la fiabilidad del XPFCP propuesto en la tesis bajo distintas condiciones de observación, parece interesante hacer uso de su flexibilidad para aplicarlo a una serie de áreas de investigación de especial interés:

- La primera de ellas es el tracking en un espacio de observación multisensorial. Para aplicar el XPFCP en este espacio de observación, bastaría con elegir la función de verosimilitud usada en el cálculo de los pesos en función del tipo de datos (es decir, sistema de observación) a integrar en el proceso de seguimiento. De este modo, el XPFCP actuaría, además, de algoritmo de fusión. El surgimiento en los últimos años de gran cantidad de trabajos sobre sistemas de seguimiento multisensorial, fundamentalmente en los llamados “*espacios inteligentes*” (en los que el XPFCP demuestra su validez a través de los resultados incluidos en el capítulo 6), como [Odobez06], justifican el interés de esta línea de trabajo.

- En el mismo área de robótica personal en “*espacios inteligentes*”, y más concretamente en aplicaciones multi-robot, la incorporación de un proceso de identificación de cada robot es inmediata a partir del XPFCP propuesto en la tesis, utilizando la odometría generada por cada robot en la ponderación del set S_t . Este proceso eliminaría del sistema de tracking las hipótesis con dinámica no coincidente con la odometría de ningún robot, y validaría e identificaría aquellos que sí coincidiesen. Este sistema se encuentra ya en fase de desarrollo a partir del trabajo [Pizarro08], realizado en el marco de esta tesis.
- Otra área de gran interés en robótica personal, fundamentalmente en este caso en el tracking con visión, es el seguimiento de personas, área en la que se centran muchos de los trabajos analizados en el capítulo 2, como [Okuma04] o [Smith05]. La adaptación del clasificador de medidas $Y_t \Rightarrow G_{i:k,t|in}$ propuesto en esta tesis a tareas específicas de detección de personas (en función de la forma, el color de la piel, etc.) así como la modificación de las componentes del vector de estado a otras más adecuadas para este caso (en función de las características usadas en la detección) serían reformas suficientes para poder utilizar el XPFCP en esta nueva aplicación, de gran interés en áreas de investigación que incluye no sólo la robótica personal sino también otras como la seguridad vial (como muestran trabajos como [Fernández08]).

Además de las líneas de aplicación del XPFCP comentadas, se abren también nuevas líneas de investigación con el desarrollo de la tesis, algunas de las cuales se listan a continuación:

- El problema más importante encontrado en las pruebas de tracking realizadas con el XPFCP es su comportamiento en situaciones de multitud. En este tipo de situaciones el seguimiento individual de cada objeto es difícil de resolver ([Reisman04]), por lo que una solución sería informar a la salida del tracking de la aparición de esta situación (que se detecta fácilmente al encontrar múltiples soluciones del proceso de asociación) y detener el lazo de ejecución del seguidor, hasta que ésta desaparezca.
- La posibilidad de utilizar modelos no lineales en el XPFCP abre la posibilidad de investigar en el diseño de una versión no lineal de este algoritmo de seguimiento multi-objeto propuesto, con múltiples aplicaciones como las indicadas en párrafos anteriores. El diseño de una versión “*Rao-Blackwellized*” (ver en el apéndice I) del XPFCP propuesto para sistemas de modelo más complejo que el usado en esta tesis y divisibles en una parte integrable y otra no, puede ser también una línea de investigación interesante ante la necesidad de usar la propuesta de tracking aquí incluida con otro tipo de modelos de estado.
- Dejando al margen la especificación de tiempo real planteada en la tesis, sería también interesante investigar en las mejoras que pueden conseguirse con la incorporación en el XPFCP de algunos de los procesos propuestos en este sentido por la comunidad científica y estudiados en esta memoria: la inserción de la transformación SUT ([Merwe01]) al PF del XPFCP, permite obtener una mejor representación de la función de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1:t})$, y por ende una ponderación del set S_t más fiable, en caso de que al aplicar el XPFCP a sistemas más complejos, la usada en la propuesta “*Bootstrap*”

([Gordon93]) no presente la robustez necesaria; por otro lado, la incorporación de la propuesta de observación basada en blobs, característica del algoritmo “*BraMBle*” ([Isard01]) puede ser interesante en la aplicación del XPFCP al tracking de múltiples objetos en “*entornos inteligentes*”.

- Finalmente, cabe destacar que el XPFCP diseñado en esta tesis presenta una estructura fácilmente exportable a sistemas de programación hardware, tanto por el flujograma del PF como por el de las estructuras de clasificación que incluye. Este tipo de implementación reporta ventajas que incluyen, no sólo el t_{exe} del algoritmo que se disminuye en gran medida, sino también el coste final del sistema de seguimiento ([Bravo07]).

APÉNDICE I

VERSIONES DEL FILTRO DE PARTÍCULAS

En este apéndice se van a analizar las variantes del PF que al nombre propio no se identifican únicamente como modificaciones del algoritmo “*Bootstrap*” que permitan resolver alguno de los problemas que aparecen en la utilización de este estimador.

Junto al análisis matemático de cada una de las alternativas presentadas en este anexo, se incluye un breve estudio de aplicabilidad y conveniencia de uso de cada propuesta al objetivo de seguimiento de múltiples objetos planteado en esta tesis.

Para comprender completamente el contenido teórico de este apéndice conviene revisar previamente los conceptos básicos sobre la funcionalidad del PF incluidos en el capítulo 4.

I.1. Filtros de partículas normalizados o regularizados

Al margen de las analizadas en el capítulo 4, una de las soluciones planteadas por la comunidad científica para resolver el problema del empobrecimiento del set S_t que caracteriza a la creencia generada como salida del PF $p(\bar{x}_t | \bar{y}_{1:t})$ básico, consiste en eliminar la etapa de selección que genera este problema de su flujograma funcional. Estas versiones del PF se asemejan, por tanto, al algoritmo SIS básico que no presenta el problema de empobrecimiento explicado al no incluir el paso de re-muestreo que lo provoca.

Por otro lado, la incorporación de la etapa de resampling al algoritmo SIS es, tal y como ya se ha comentado, imprescindible para evitar la degeneración del estimador. Para compensar la falta de este paso en las versiones del PF descritas en este punto, se incorpora una etapa de normalización consistente en obtener una función continua tras el paso de corrección $p(\hat{x}_t | \bar{y}_{1:t})$, que será la que represente finalmente al valor de la creencia estimado por el filtro.

La siguiente iteración del PF así modificado comienza, por tanto, con un proceso de muestreo del que se obtiene el set de partículas \hat{S}_{t-1} que caracterice al valor de partida de la creencia $p(\hat{x}_{t-1} | \bar{y}_{1:t-1})$ en esa siguiente iteración.

I.1.1. “Gaussian Particle Filter” (GPF) y “Gaussian Sum Particle Filter” (GSPF)

En [Kotecha03] se describe una versión alternativa al algoritmo SIR muy extendida entre la comunidad científica para resolver el problema de empobrecimiento que este sufre. Se trata del “Filtro de Partículas Gaussiano” (“Gaussian Particle Filter”, GPF).

Como se explica en los párrafos anteriores, el empobrecimiento del conjunto de partículas de salida S_t no se produce en el GPF, al eliminar de su flujograma de funcionamiento la etapa de resampling del SIR que lo produce. En el GPF, esta etapa se sustituye por otra de generación de una función continua $p(\hat{x}_t | \bar{y}_{1:t})$ que caracteriza al valor final de la creencia, y que da nombre al GPF, al tratarse de una distribución gaussiana.

Para obtener esta función $p(\hat{x}_t | \bar{y}_{1:t})$ continua es necesario considerar que el set de muestras S_t' a la salida del paso de corrección puede representarse mediante una densidad unimodal gaussiana ($p(\hat{x}_t | \bar{y}_{1:t}) = \text{norm}(N_t, \Omega_t)$), y obtener sus estadísticos de media N_t y varianza Ω_t^2 .

La Figura I.1 presenta a modo de ejemplo, el funcionamiento en una iteración de un GPF en una aplicación de vector de estado x_t unidimensional.

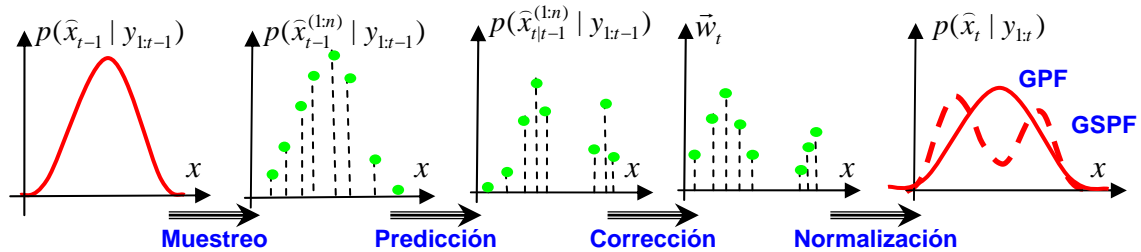


Figura I.1. Representación funcional del GPF para la estimación de una variable unidimensional. En el paso de normalización se muestra, a modo de ejemplo, el valor estimado de la creencia $p(\hat{x}_t | y_{1:t})$ que arrojaría un GPF (en línea continua) y un GSPF (en línea discontinua).

Tal y como se desprende de los párrafos anteriores y se observa en la Figura I.1, el inconveniente principal del uso de un GPF como estimador se debe a la aproximación de la

distribución caracterizada por el set de partículas S'_t , mediante una función unimodal gaussiana $p(\hat{x}_t | \bar{y}_{1:t})$. Debido a ello el GPF presenta problemas de exactitud y solo se puede utilizar en aplicaciones de estimación unimodal, con lo que su uso queda descartado en la aplicación de seguimiento planteada en la tesis, al no cumplir las especificaciones establecidas.

Existe una versión mejorada del GPF, el “*Filtro de Partículas de Suma de Gaussianas*” (“*Gaussian Sum Particle Filter*”, GSPF). En esta versión la creencia continua aproximada $p(\hat{x}_t | \bar{y}_{1:t})$ es normalizada mediante la suma de un número variable l de gaussianas, en lugar de utilizar una única función de este tipo, como se muestra en la siguiente expresión:

$$p(\hat{x}_t | \bar{y}_t) = \sum_{j=1}^l \text{norm}(N_{j,t}, \Omega_{j,t}) \quad \langle \text{I.1} \rangle$$

Esta sencilla modificación elimina la problemática asociada al uso del GPF para desarrollar tareas de estimación multimodales como la abordada en la tesis, si bien incrementa de forma notable la carga computacional del algoritmo.

Finalmente, conviene destacar que el GSPF es una particularización del RPF, cuya descripción se aborda en el apartado siguiente, por lo que se aconseja leer los siguientes párrafos para obtener un conocimiento más detallado de este estimador.

I.1.2. “*Regularized Particle Filter*” (RPF)

Musso y Oudjane, exponen en [Doucet01] una última versión del estándar SIR que merece la pena comentar en este punto: el “*Filtro de Partículas Regularizado*” (“*Regularized Particle Filter*”, RPF). La innovación incluida en el RPF respecto al PF básico es la inclusión de un paso de regularización del conjunto de partículas, con el objetivo de asegurar la diversidad del set S_t , de forma semejante a la etapa de normalización característica del GPF.

Los autores proponen dos alternativas de regularización: antes del proceso de corrección (“*Pre-RPF*”) o después de éste (“*Post-RPF*”).

La etapa de regularización consiste en obtener una función continua $g(S) \Rightarrow p(\hat{x} | \bar{y})$ a aplicar al conjunto de partículas $(S_{t|t-1} = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_{t-1}^{(i)}\}_{i=1}^n$ en el caso del “*Pre-RPF*”, o $S'_t = \{\bar{x}_{t|t-1}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^n$ en el del “*Post-RPF*”) para obtener una PDF continua $p(\hat{x}_t | \bar{y}_{1:t})$ que caracterice la creencia a la salida del PF, y que es muestreada en la etapa de predicción de la siguiente iteración del filtro. De este modo, al igual que en el caso del GPF, se evita la ejecución del paso de re-muestreo del PF estándar, causante del empobrecimiento set de partículas S_t .

La Figura I.2, extraída de [Doucet01], presenta un ejemplo unidimensional del modo de obtener la función continua de regularización del set $g(S) \Rightarrow p(\hat{x} | \bar{y})$, a partir del valor y el peso del conjunto de partículas $S = \{x^{(i)}, \tilde{w}^{(i)}\}_{i=1}^n$.

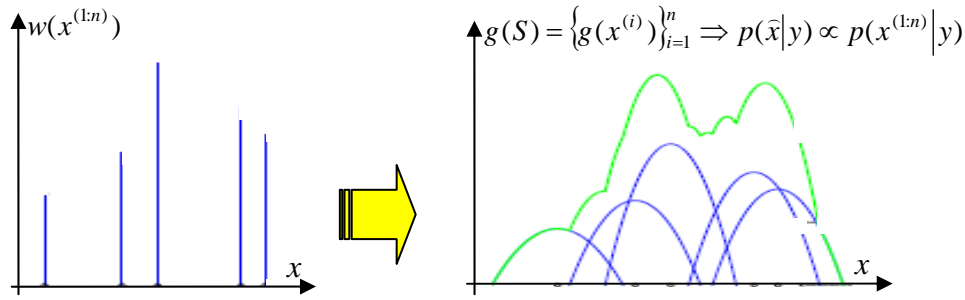


Figura I.2. Representación funcional de la obtención de la función continua de regularización del set de partículas $g(S)$ característica del RPF, para un problema de estimación unidimensional. Extraída de [Doucet01].

La función de regularización $g(S)$ se obtiene (como se aprecia en la Figura I.2) sumando las funciones continuas asociadas a cada partícula del set $S = \{x^{(i)}, \tilde{w}^{(i)}\}_{i=1}^n$. De forma analítica:

$$g(S) = \{g(\tilde{x}^{(i)})\}_{i=1}^n \Rightarrow p(\tilde{x}|y) \propto p(x^{(1:n)}|y) \quad <I.2>$$

El RPF tiene una funcionalidad muy semejante a la del GPF descrito anteriormente. La característica principal que diferencia al RPF del GPF se encuentra en la PDF continua que representa a la creencia $p(\tilde{x}_t | \tilde{y}_{1:t})$ a la salida del filtro, pues en el caso del GPF esta función ha de ser gaussiana $p(\tilde{x}_t | \tilde{y}_{1:t}) = norm(N_t, \Omega_t)$ (o suma de gaussianas en el caso del GSPF $p(\tilde{x}_t | \tilde{y}_{1:t}) = norm(N_t, \Omega_t)$), mientras que en el RPF no existe restricción alguna para ésta.

Debido a este hecho, la ejecución del GPF conlleva menor carga computacional que el RPF, pues en el caso del GPF solo es necesario obtener el valor de los estadísticos de media N_t y varianza Ω_t^2 que caracterizan a la función $p(\tilde{x}_t | \tilde{y}_{1:t})$, mientras que en el caso del RPF es necesario trabajar con la función completa $p(\tilde{x}_t | \tilde{y}_{1:t})$. En contrapartida, el RPF puede modelar densidades de creencia más complejas que las generadas por el GPF o el GSPF.

Los autores del RPF, Musso y Oudjane, presentan en [Doucet01] varias propuestas para la función de regularización $g(\tilde{x}^{(i)})$ a asociar a cada partícula $\tilde{x}^{(i)}$, teniendo como objetivo en todos los casos la minimización del error entre la función discreta representada por el set sin regularizar, $S = \{x^{(i)}, \tilde{w}^{(i)}\}_{i=1}^n \Rightarrow p(\tilde{x}^{(1:n)} | \tilde{y})$, y la continua obtenida tras la regularización del mismo $p(\tilde{x} | \tilde{y})$:

- La propuesta más sencilla es usar como función de regularización una gaussiana asociada a cada partícula del set, $g(\tilde{x}^{(i)}) = norm(N_i, \Omega_i)$. Esta es la opción utilizada en el ejemplo mostrado en la Figura I.2. En este caso, el RPF se comporta como un GSPF, pues en ambos

casos la función continua que caracteriza a la creencia $p(\hat{x}_t | \bar{y}_{1:t})$ a la salida del estimador resulta ser la suma de las funciones gaussianas asociadas a las partículas del conjunto $S = \{x^{(i)}, \tilde{w}^{(i)}\}_{i=1}^n$. Desde un punto de vista analítico:

$$p_{GSPF}(\hat{x}_t | \bar{y}_t) \equiv p_{RPF}(\hat{x}_t | \bar{y}_t) = \sum_{i=1}^n \text{norm}(N_{i,t}, \Omega_{i,t}) \quad \langle I.3 \rangle$$

- La función de regularización óptima $g(S) \Rightarrow p(\bar{x} | \bar{y})$ es, según los autores del RPF, el denominado núcleo de Epanechnikov. Sin embargo, el uso de esta función no es recomendable en aplicaciones de tiempo real, debido al alto t_{exe} que lleva su cálculo.

La Figura I.3 muestra un diagrama funcional de las dos versiones del RPF comentadas. En la figura se destaca el carácter continuo o discreto de la PDF que representa a la estimación en cada etapa del RPF, y la funcionalidad y ubicación del proceso de regularización en el flujograma de esta versión del PF.

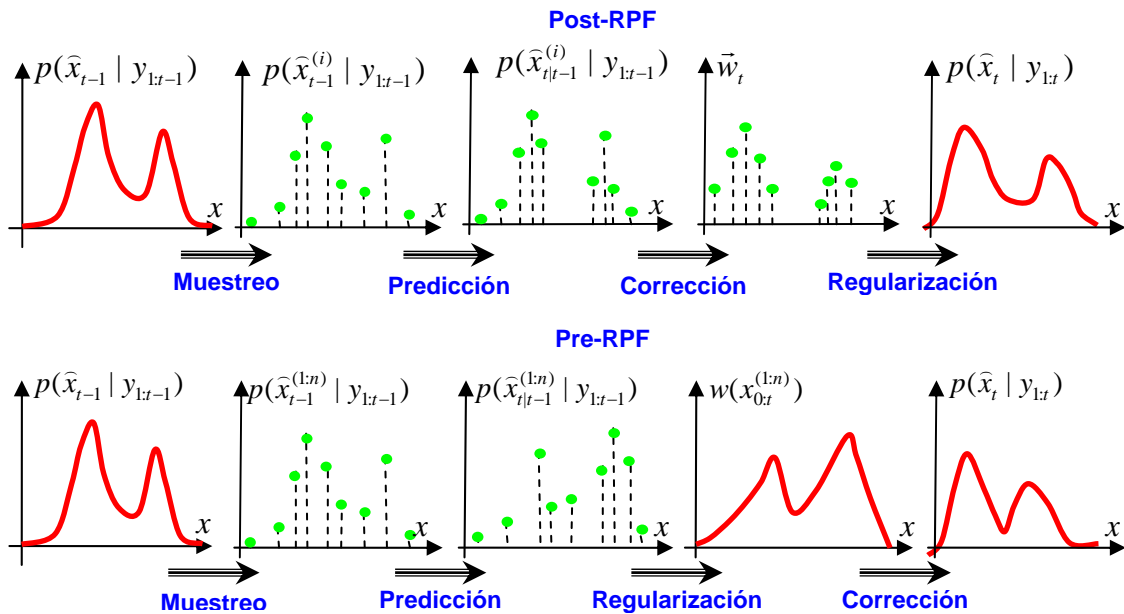


Figura I.3. Representación funcional de los algoritmos “Pre-RPF” y “Post-RPF” para la estimación de una variable unidimensional.

Del análisis de la Figura I.3 se desprende que la versión “Post-RPF” implica una mayor complejidad computacional que la versión “Pre-RPF”, debido a que la ejecución de la primera de las ellas conlleva el cálculo de la forma continua de la función de ponderación, $w(x_{0:t}^{(1:n)})$.

I.2. El algoritmo “BramBLE”

El nombre del algoritmo “BramBLE” viene de “Bayesian Multiple-BLob tracker”, es decir, seguidor bayesiano de múltiples blobs. Son muchas las aplicaciones de visión en las que el

trabajo con blobs como elementos indivisibles de medida es un hecho ya habitual, debido sobre todo a la opción cada vez más común de modelar la imagen mediante modelos de apariencias.

El algoritmo “*BraMBLe*”, introducido en [Isard01], es un PF estándar, cuya característica principal es definir el vector de medidas, y, por tanto, la verosimilitud, mediante blobs.

Gracias a este procedimiento los modelos implicados en las ecuaciones del estimador se simplifican en gran medida, lo que conlleva la disminución de la carga computacional del sistema de seguimiento final. Además el uso de modelos basados en blobs incrementa la robustez del tracking pues éstos son estables ante cambios de las condiciones de entorno.

El algoritmo “*BraMBLe*”, se utiliza además como proceso de detección de nuevos objetos en la imagen, en aplicaciones en las que el fondo es constante (sobre todo vigilancia). El blob que caracteriza al fondo constante de la imagen se incluye en el estimador bayesiano como un target más, obteniendo de este modo una predicción del mismo para el siguiente paso de ejecución. Si no existe correspondencia en esta estimación en algún punto de la imagen se concluye que se debe a la aparición de un nuevo target en la escena. De esta forma se elimina el proceso de sustracción de fondo típico en estas aplicaciones que no presenta gran fiabilidad ([Isard01]).

Finalmente, en el vector de estado a obtener por el PF también se incluye información acerca de la interacción de los distintos objetos de la escena, a través de una variable que indica a grandes rasgos la posición en profundidad del objeto en la imagen. De este modo, el sistema pueda manejar situaciones de oclusión parcial o total de los elementos seguidos.

Sin embargo, la inclusión de las variables comentadas en los dos párrafos anteriores conlleva una complicación del modelo de estado a estimar, que además, será de dimensiones variables si se utiliza en aplicaciones de seguimiento de un número cambiante de targets. La simplificación del algoritmo “*BraMBLe*” conseguida gracias al uso de modelos en base a blobs se pierde de este modo, resultando que en aplicaciones con restricciones de tiempo real, el seguidor no se puede utilizar para el tracking de más de tres o cuatro objetos ([Tweed02]).

En cualquier caso, el algoritmo “*BraMBLe*” o alguna de las ideas básicas que implica es utilizado en trabajos posteriores (como [Branson05]) y sirve de continua referencia en otros muchos, debido que reúnen un rigor matemático que los hace exactos en la tarea de seguimiento y la simplicidad a la hora de resolver problemas de detección, asociación y tracking de varios objetos a partir de información de visión. En el capítulo 2 se analizan algunos de estos trabajos, para comparar sus resultados con los obtenidos con el sistema de MTT propuesto en esta tesis.

I.3. “*Rao-Blackwellized Particle Filter*” (RBPF)

Los “*Filtros de Partículas Rao-Blackwellized*” o RBPFs (“*Rao-Blackwellized Particle Filters*”) tienen su origen en la aplicación del teorema de “*Rao-Blackwell*” ([Doucet01]) al proceso de estimación probabilística expuesta en este documento. La idea central del teorema es que cualquier estimador, independientemente de su eficiencia, puede optimizarse según el criterio del error cuadrático mínimo si se transforma de forma adecuada el modelo del propio estimador o del vector de estado a estimar.

Este teorema se encuentra aplicado a problemas de seguimiento de objetos con PF en múltiples casos, tal y como prueban varias de las referencias analizadas en el capítulo 2 ([Torma03], [Schulz03a], [Khan04] o [Sarkka07]). El objetivo del uso de este teorema, aplicado al PF en su tarea de estimación multimodal, es simplificar las dimensiones del vector de estados implicado en el proceso recursivo de estimación bayesiano, y por ende, disminuir: el t_{exe} del seguidor, que de otro modo no podría ejecutarse en tiempo real en algunos casos ([Khan04]); e indirectamente, la tasa de fallos de la estimación ([Dellaert03]).

El procedimiento básico en que consiste la aplicación del teorema de “*Rao-Blackwell*” al desarrollo de un PF es la partición del vector de estado $\bar{x}_{0:t} = \{\bar{x}_{1,0:t}, \bar{x}_{2,0:t}\}$ en la densidad de creencia $p(\bar{x}_{0:t} | \bar{y}_{1:t})$ a través de la regla de Bayes, donde:

$$p(\bar{x}_{0:t} | \bar{y}_{1:t}) = p(\{\bar{x}_{1,0:t}, \bar{x}_{2,0:t}\} | \bar{y}_{1:t}) = p(\bar{x}_{1,0:t} | \bar{y}_{1:t}) \cdot p(\bar{x}_{2,0:t} | \bar{y}_{1:t}), \quad \langle I.4 \rangle$$

- el cálculo de $p(\bar{x}_{1,0:t} | \bar{y}_{1:t})$ no tiene una solución analítica integrable a través de la ecuación compacta de Bayes <2.15>, expuesta en el capítulo 2,
- y $p(\bar{x}_{2,0:t} | \bar{y}_{1:t})$ puede obtenerse de forma analítica a través de dicha ecuación.

A partir de la expresión distribuida de la creencia mostrada en <I.4>, por un lado se calcula la parte integrable de esta PDF $p(\bar{x}_{2,0:t} | \bar{y}_{1:t})$, relacionada con la parte $\bar{x}_{2,0:t}$ del modelo de de estados, y se estima por otro mediante un PF, la parte no integrable de la misma $p(\bar{x}_{1,0:t} | \bar{y}_{1:t})$, representada en este caso por un conjunto de partículas $S_{1,t} = \{\bar{s}_{1,i,t} = \bar{x}_{1,t}^{(i)} / i = 1:n\}$, para obtener el valor final de la creencia $p(\bar{x}_{0:t} | \bar{y}_{1:t})$, o de su valor marginal $p(\bar{x}_t | \bar{y}_{1:t})$.

En la mayor parte de los casos en los que se utiliza la versión “*Rao-Blackwellized*” del PF, se suele usar un KF para obtener la parte integrable de la creencia $p(\bar{x}_{2,0:t} | \bar{y}_{1:t})$ (pues suele corresponderse con un proceso gaussiano y cuasi-lineal), mientras que el PF que se utiliza para calcular la parte no integrable de la misma $p(\bar{x}_{1,0:t} | \bar{y}_{1:t})$.

Con esta modificación, el número de partículas necesarias para su funcionamiento eficiente se ve disminuido en algunos casos en hasta un 80% ([Torma03]).

Para poder desarrollar la expresión <I.4> se parte del hecho de que las dos partes en que se divide el vector de estado $\bar{x}_{0:t} = \{\bar{x}_{1,0:t}, \bar{x}_{2,0:t}\}$ incluyen componentes independientes. Si este hecho no se cumple, la ecuación <I.4> tiene que reescribirse de la siguiente forma:

$$p(\bar{x}_{0:t} | \bar{y}_{1:t}) = p(\bar{x}_{1,0:t}, \bar{x}_{2,0:t} | \bar{y}_{1:t}) = p(\bar{x}_{1,0:t}^{(i)} | \bar{x}_{2,0:t}, \bar{y}_{1:t}) \cdot p(\bar{x}_{2,0:t} | \bar{y}_{1:t}), \quad \langle I.5 \rangle$$

donde se ha desarrollado la densidad de creencia $p(\bar{x}_{0:t} | \bar{y}_{1:t})$ de modo que la parte independiente del vector de estados sea $\bar{x}_{2,0:t}$, es decir, la relacionada con la parte integrable de

la creencia. En este caso será necesario calcular de forma analítica $p(\bar{x}_{2,0t} | \bar{y}_{1t})$ en primer lugar, y a partir de este valor obtener $p(\bar{x}_{1,0t} | \bar{x}_{2,0t}, \bar{y}_{1t})$.

Sin embargo, en caso de que la variable independiente sea la parte del vector de estados $\bar{x}_{1,0t}$ relacionada con la PDF no integrable de la creencia, la ecuación <I.4> quedará así:

$$p(\bar{x}_{0t} | \bar{y}_{1t}) = p(\bar{x}_{2,0t}, \bar{x}_{1,0t} | \bar{y}_{1t}) = p(\bar{x}_{2,0t} | \bar{x}_{1,0t}, \bar{y}_{1t}) \cdot p(\bar{x}_{1,0t} | \bar{y}_{1t}) \quad <I.6>$$

En este caso será necesario calcular la parte integrable $p(\bar{x}_{2,0t} | \bar{x}_{1,0t}, \bar{y}_{1t})$ para cada $\bar{x}_{1,0t}^{(i)}$, es decir, para cada partícula del PF con el que se obtiene la parte independiente de la creencia $p(\bar{x}_{1,0t}^{(i)} | \bar{y}_{1t})$.

La situación comentada en el párrafo anterior es típica en aplicaciones de seguimiento de múltiples objetos, como el tratado en esta tesis, en el caso en que se introduzca dentro del modelo de estados a estimar una variable $\theta_{i,j}$ que describa la asociación de cada objeto i con el resto $j = 1:k \notin l$ (como en [Schulz03a]). En estas circunstancias, la versión “*Rao-Blackwellized*” del PF no alcanza la eficiencia comentada en párrafos anteriores ([Doucet01]) puesto que la carga computacional del proceso global crece al tener que calcular la parte integrable de la creencia $p(\bar{x}_{2,0t} | \bar{x}_{1,0t}^{(i)}, \bar{y}_{1t})$ para cada partícula $\bar{x}_{1,t}^{(i)}$ que indicará en este caso cada posible relación de asociación $\theta_{i,j}$. El número de partículas del PF utilizado para calcular la parte no integrable de la creencia $p(\bar{x}_{1,0t}^{(i)} | \bar{y}_{1t})$ es, en evidentemente este caso, un factor determinante en el t_{exe} del estimador global.

Por otro lado, algunos autores (ver en [Doucet01]) rechazan el uso de los RBPFs debido a su baja robustez. Si por alguna razón cambia el comportamiento del sistema, introduciéndose una no-linealidad en la parte integrable de la creencia, la propuesta diverge rápidamente, sobre todo en el caso (ver la expresión <I.5>) de que $\bar{x}_{1,t}^{(i)}$ (relacionado con la parte no integrable de la creencia $p(\bar{x}_{1,0t}^{(i)} | \bar{y}_{1t})$) dependa de $\bar{x}_{2,0t}$ (relacionado con la parte integrable $p(\bar{x}_{2,0t} | \bar{y}_{1t})$).

I.4. “*Unscented Particle Filter*” (UPF)

El “*Filtro de Partículas Unscented*” (“*Unscented Particle Filter*”, UPF) es una versión del PF basada en la inserción de un filtro UKF en el PF que permita obtener una mejor función de aproximación a la creencia $q(\bar{x}_t | \bar{x}_{0:t-1}, \bar{y}_{1t})$.

El modo de funcionamiento del UPF es idéntico al del EKPF explicado en el capítulo 4, siendo la única diferencia el método de cálculo de la función de aproximación individual, $p(\bar{x}_t^{(i)} | \bar{x}_{0:t-1}^{(i)}, \bar{y}_{1t})$, pues utiliza para ello un UKF.

Desde un punto de vista analítico, el UKF tiene la misma funcionalidad y flujograma que el KF. Sin embargo las PDFs implicadas en el proceso de estimación del UKF añaden estadísticos de segundo orden a los de primer orden, característicos de las densidades implicadas en un KF básico. La forma de obtener estos parámetros secundarios es a través de una “*Scaled Unscented Transformation*” (SUT).

El desarrollo analítico de la transformación SUT, que no se incluye en este documento, puede ser consultado en [Julier97] o [Merwe01]. En este punto merece la pena destacar a este respecto únicamente el hecho de que la transformación SUT permite extraer los estadísticos de segundo orden que definan a la PDF $p(\bar{x}_t^{(i)} | \bar{x}_{0:t-1}^{(i)}, \bar{y}_{1:t})$ a través de los llamados “*Puntos Sigma*” (“*Sigma Points*”, SPs), que son un conjunto de muestras ponderadas que conforman un set semejante al de partículas del PF, distribuidos aleatoriamente para representar a la densidad de aproximación a la creencia individual asociada a cada partícula.

Del párrafo anterior se puede extraer una idea acerca de la complejidad computacional del UPF, pues la etapa de predicción del UPF incluye, para cada partícula del conjunto de entrada $(\bar{s}_{i,t-1} = \bar{x}_{i,t-1}^{(i)} / i = 1 : n)$, los siguientes pasos:

- Cálculo del set de SPs mediante la aplicación de la transformación SUT.
- Actualización de cada uno de los SPs mediante las ecuaciones del KF modificadas del UKF. Este paso implica el cálculo de los valores de media $N_{i,t}$ y varianza $\Omega_{i,t}$ del set de SPs, que caracterizarán a la partícula.
- Obtención de la función gaussiana que representa a la partícula $(p(\bar{x}_t^{(i)} | \bar{x}_{0:t-1}^{(i)}, \bar{y}_{1:t}) = \text{norm}(N_{i,t}, \Omega_{i,t}))$ a partir de los estadísticos de media y varianza obtenidos del set de SPs.
- Muestreo de la función $p(\bar{x}_t^{(i)} | \bar{x}_{0:t-1}^{(i)}, \bar{y}_{1:t})$ para obtener un nuevo valor de partícula $(\hat{s}_{i,t-1} = \hat{x}_{i,t-1}^{(i)} / i = 1 : n)$.

Existen pocos trabajos publicados en los que el UPF se utilice como proceso de seguimiento a partir de información de visión, debido tal y como ya se ha comentado, al alto coste computacional del algoritmo. En [Rui01] se presenta un ejemplo de aplicación del UPF a la tarea de tracking expuesta. Los autores del artículo justifican el uso de este algoritmo en su objetivo de evitar el problema de degeneración que aparece al utilizar el PF estándar bajo ciertas condiciones del sistema, y que ya han sido explicadas en el capítulo 4. En [Rui01] se presentan resultados arrojados por el UPF en el seguimiento de pequeños movimientos de un único objeto, dentro de un entorno muy reducido.

Sin embargo, la autora de la tesis no ha encontrado ninguna referencia importante de aplicación de un UPF a una tarea de MTT, como la que se aborda en el presente trabajo.

APÉNDICE II

CONSIDERACIONES SOBRE LOS PROCESOS DE CLASIFICACIÓN

En este apéndice se incluyen varias consideraciones sobre los clasificadores, que, si bien no constituyen una aportación propia de la tesis, es necesario tenerlos en cuenta a la hora de analizar el funcionamiento de los algoritmos de segmentación propuestos.

Este capítulo constituye, por tanto, un anexo a los contenidos expuestos en el capítulo 5, por lo que se recuerda aquí la nomenclatura relacionada con el proceso de segmentación, cuya base es la definición de conjunto de k clases $G_{1:k}$ que se obtienen como resultado del clasificador:

$$G_{j=1:k} \equiv \left\{ \vec{g}_j, \vec{l}_{1:l_j, j} \right\}_{j=1}^k = \left\{ \vec{g}_j, L_j \right\}_{j=1}^k, \quad \langle \text{II.1} \rangle$$

donde $L_j = \left\{ \vec{l}_{i,j} / i = 1 : l_j \right\}$ referencia al set de l_j miembros que la clasificación asocia al cluster j . Además el set de datos a organizar, del que L_j es un subconjunto, se va a referenciar a través de la matriz de datos $Y = \left\{ \vec{y}_i / i = 1 : m \right\}$. Por otro lado \vec{g}_j es el vector característico que identifica al cluster j , que en general recibe el nombre de *centroide*¹.

¹ Ver el capítulo 5 para una definición más completa de este concepto de “*centroide*”.

II.1. Características de los algoritmos de clasificación

Como regla general, los distintos métodos de clasificación de datos se caracterizan por las siguientes propiedades básicas ([Jain99]):

- *Recursividad.* Muchos segmentadores, como el “*k-medias*”, son recursivos, es decir, se ejecutan de forma iterativa hasta que se encuentra una solución que satisface cierta especificación. El inconveniente principal de la recursividad es que el proceso puede no converger y no alcanzarse nunca la condición que finaliza su ejecución. Este tipo de algoritmos tienen, por tanto, un t_{exe} variable pero normalmente bajo, si convergen, cuestión importante en la aplicación de interés de MTT de la tesis.
- *Repetibilidad.* Un clasificador será repetible si la organización que arroja como resultado de su ejecución no depende del orden de entrada de los datos al éste. Esta característica, que no es propia de todos los métodos de segmentación (no lo es por ejemplo de los algoritmos jerárquicos, pero sí de los dos propuestos en la tesis) es importante en aplicaciones como la de interés en la que los datos no presentan un orden determinado antes de ser clasificados.
- *La forma de proporcionar la solución.* La mayor parte de métodos de clasificación generan como resultado la lista de los k clusters encontrados en la base de datos ($G_j / j=1:k$), identificados cada una de ellos por un elemento representativo (en general llamado centroide, \bar{g}_j). Si además el proceso proporciona junto al identificador la lista de miembros que se asocia a cada grupo ($L_j = \{i_{i,j} / i=1:l_j\}$) (como en el caso del “*k-medias*”) se dice que se trata de un método “*linkage*” o con conexión, mientras que en caso contrario (“*subtractive*”) se tratará de un proceso “*no-linkage*” o sin conexión. Esta característica es también importante en la aplicación de interés, pues los datos asociados a cada clase son utilizados en el PF para robustecer la creencia relacionada con cada objeto, como se explica en el capítulo 3.
- *Clasificación “todo/nada” (“booleana”) o “borrosa”.* El grado de pertenencia de cada dato a una o más clases determinará si la clasificación de los datos es: borrosa cuando un dato puede pertenecer a una clase con un grado de pertenencia entre 0 y 1, normalizando la variable pertenencia; o booleana si el grado de pertenencia de cada dato sólo puede ser 0 ó 1. Generalmente la definición de los algoritmos borrosos implica también la posibilidad de que un dato se asocie a varios grupos con distintos grados de pertenencia, o incluso que no se asocie a ninguno; mientras que el carácter booleano de los algoritmos todo/nada (como el “*k-medias*”) implica que cada dato debe pertenecer necesariamente a uno y solamente a un cluster. Desde este punto de vista, los más adecuados para la aplicación de interés son los borrosos, como el “*subtractive*”, pues permiten manejar con mayor fiabilidad los ruidos en las medidas, como se expone en el capítulo 5. En la literatura inglesa a la organización borrosa se le denomina blanda (“*soft*”), y a la booleana dura (“*hard*”).
- *Número de clases conocido a priori.* Muchos de los segmentadores propuestos en la literatura científica, como el “*k-medias*”, precisan de la especificación, como parámetro de

entrada, del número de clases k en las que se desea organizar el conjunto Y . Será necesario modificar estos procesos para eliminar esta característica de su funcionalidad si se desea utilizarlos en la aplicación de interés, en la que se especifica que el número de objetos en el entorno es desconocido a priori y además variables con el tiempo.

Las diferencias que presentan los algoritmos “ k -medias” y “*subtractive*” en las características expuestas, les harán más o menos adecuados para la aplicación de MTT de la tesis en distintos aspectos que son analizados a lo largo del capítulo 5.

II.2. Tipos de algoritmos de clasificación

A partir de las características comentadas en el apartado anterior, una revisión bastante completa de las alternativas de segmentación ideadas por la comunidad científica se recoge en [Kainulainen02] y se muestra en la Figura II.1.

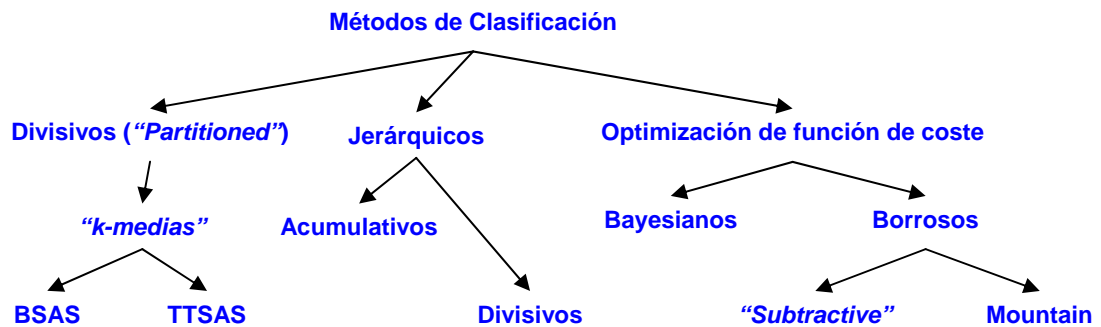


Figura II.1. Métodos de clasificación. Extraído de [Kainulainen02].

Como se aprecia en la Figura II.1, la organización que generalmente se expone en la literatura especializada define principalmente tres tipos de clasificadores:

1. *Divisivos ("Partitioned")*: Se denominan así por su carácter booleano en la forma de definir la función de pertenencia de los datos a cada grupo, pues se caracterizan por asignar cada dato a uno único. En general el número de grupos k en las que se organiza el set de datos debe ser conocido a priori. Además los algoritmos divisivos suelen ser recursivos. El algoritmo “ k -medias” pertenece a este tipo de clasificadores. Se utiliza, sobre todo, cuando el número de datos a ordenar m es alto, pues el resto de algoritmos no son aplicables en estos casos debido a su alto t_{exe} ([Pelleg00]). La Figura II.2, extraída de [Jain99], expone el resultado de una segmentación divisiva de un conjunto de datos definidos en un espacio bidimensional ($\lambda_T = 2$) en tres clases. En la figura, cada dato ha sido etiquetado con el número de la clase al que ha sido asociado por el clasificador. Se han añadido unas líneas imaginarias que dividen el espacio de características usado como, aparentemente, lo ha hecho el clasificador. El carácter booleano de la segmentación se evidencia claramente en la figura, los datos son repartidos entre los tres clusters de modo que cada uno de ellos se asocia a un sólo cluster.

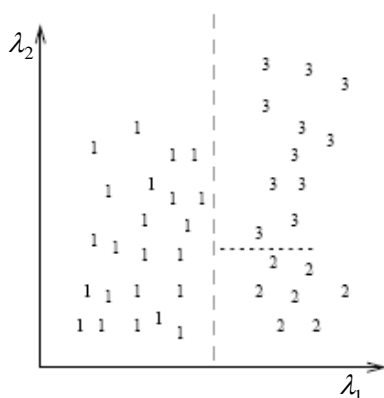


Figura II.2. Ejemplo de clasificación divisiva de un conjunto de datos definidos en un espacio bidimensional, extraído de [Jain99].

2. *Jerárquicos*: Se trata también de algoritmos de asignación booleana, pero no iterativos. Se caracterizan por requerir un paso previo en el proceso de clasificación, el de construcción de un árbol o grafo jerárquico de pertenencia. En función del punto desde el que se comience la creación del árbol, el proceso es “*acumulativo*”, si se parte de asignar a cada dato \vec{y}_i un nodo en el punto más bajo de la jerarquía y se van asociando éstos hasta que se alcanza la cumbre de la jerarquía en la que solo aparece un único grupo con todo el set Y ; o “*divisivo*”, si el árbol jerárquico se crea en sentido opuesto. Construido el grafo se establece en él un punto de corte en función del número de clases k que se desee obtener a la salida del clasificador. Además, las ramas del grafo suelen tener dirección y peso, de modo que se pueda calcular cuál es la agrupación que menor coste tiene en función de una serie de variables, que coincidirá con la organización óptima de los datos. Estos métodos proporcionan, por tanto, la solución óptima de segmentación, pues permiten analizar todas las soluciones posibles a partir del árbol de jerarquía. Sin embargo debido al su alto t_{exe} sólo se utilizan en ciertas aplicaciones en las que el número de datos a agrupar (m) es pequeño. Este tipo de procesos no es, por tanto, aplicable a la tarea de MTT de interés. En la Figura II.3 se expone el resultado de una segmentación jerárquica, también extraído de [Jain99]. En este caso el resultado se muestra a través de un árbol de pertenencia, y el número de clusters en que el clasificador organiza el conjunto de datos $Y = \{\vec{y}_1.. \vec{y}_7\}$ se selecciona moviendo el punto de corte en el árbol, que en la figura está marcado por una línea discontinua horizontal.
3. *Optimización de función de coste*: Este tipo de algoritmos no aparece en la organización de [Jain99], pero si en referencias posteriores como [Kainulainen02] o [Berkhin02], pues hasta hace pocos años se incluía dentro del tipo divisivo. Los métodos de segmentación a través de una densidad o función de coste basan su funcionamiento en la búsqueda de una solución de agrupación que minimice una función de coste. Debido a este modo de trabajo no requieren una especificación de número de clases k a buscar en el set Y , y en muchos casos son recursivos. Además, dentro de este tipo de procesos de clasificación se distinguen los algoritmos borrosos, como el “*subtractive*” o los bayesianos como el EM, mencionado en el capítulo 2. En la Figura II.4 se expone el resultado de la segmentación de un set de datos en un espacio bidimensional ($\lambda_T = 2$) en tres grupos, que optimiza la densidad de

datos por grupo. El histograma del conjunto Y se muestra en azul y un círculo rojo encierra a los miembros de cada clase. Como se aprecia en la figura, la organización resultante es borrosa, pues hay datos relacionados con dos clusters.

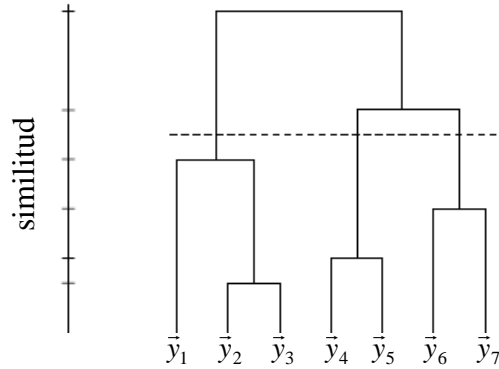


Figura II.3. Ejemplo de clasificación jerárquica extraído de [Jain99].

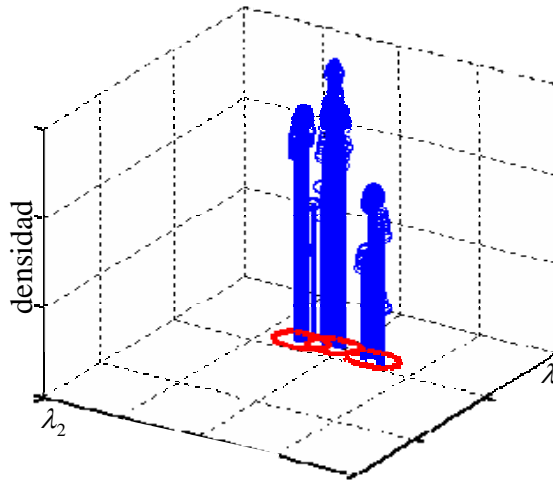


Figura II.4. Ejemplo de clasificación borrosa de un conjunto de datos definidos en un espacio bidimensional.

Al margen de los tres grandes tipos de segmentadores descritos, en [Kainualinen02] se define un cuarto tipo en el que incluye otras clases de algoritmos de entre los que merece la pena destacar los mapas auto-organizativos ([Kohonen82]), las redes neuronales [Hertz91], los algoritmos de aprendizaje competitivo y los genéticos ([Holland75]). Todos ellos son procesos que requieren un entrenamiento previo y un conocimiento a priori acerca de los grupos, tanto en número como en características, en las que se va a organizar el set Y . Se trata de algoritmos relacionados, más bien, con el área de aprendizaje de modelos (“*Indirect Mining*”), y no son, por tanto adecuados para la aplicación de MTT en la tesis.

Es conveniente tener presente que el clasificador en la aplicación final de seguimiento abordada en este documento, es una etapa previa que sirve de complemento al proceso central y más importante de estimación. Debido a ello, no es tan relevante la eficacia del segmentador sino su agilidad (bajo t_{exe}) a la hora de proporcionar información organizada al PF encargado

de la tarea de estimación, pues es esta última la que aporta la fiabilidad necesaria en el proceso global de tracking.

Como comentario final indicar que una buena regla para diseñar o elegir un segmentador eficiente es que se rija por reglas sencillas y robustas y sea fácil de implementar, pues de este modo se asegurará la rápida y fiable convergencia del mismo ([Kainulainen02]).

II.3. Formas de la medida de distancia

Tal y como se introduce en el capítulo 5, la medida de distancia en el espacio de características elegido para la clasificación, es la variable en la que más habitualmente se basa el funcionamiento de este proceso. En este apartado se revisan las formas más típicas de esta medida, justificando el uso de cada una de ellas en función de las características de la segmentación, y llegando así a la conclusión de la distancia euclídea es la más adecuada en la aplicación de interés en la tesis.

La ecuación <II.2> la describe a través de lo que se denomina en general norma d_δ en un espacio λ_T dimensional ([Jain99]):

$$d_{\pi_{i,j}} = \left(\sum_{\lambda=\lambda_1}^{\lambda_T} (a_{\lambda,i} - b_{\lambda,j})^\delta \right)^{1/\delta}, \quad \text{<II.2>}$$

donde el sumatorio permite recorrer todas las λ_T dimensiones del espacio.

Las variables $\vec{g}_j = [b_{1,j} \dots b_{\lambda_T,j}]^T$ e $\vec{y}_i = [a_{1,i} \dots a_{\lambda_T,i}]^T$ de la ecuación <II.2> se refieren generalmente al centroide \vec{g}_j del grupo j y al dato \vec{y}_i , candidato a ser miembro de ese grupo, respectivamente.

De la explicación anterior se extrae que, para poder realizar el cálculo expuesto, es necesario elegir en primer lugar un elemento característico de cada clase, su centroide \vec{g}_j , respecto al que obtener la medida de distancia de cada dato del set Y a clasificar, en el espacio definido. En los procesos de segmentación divisivos como “*k-medias*”, la elección de los centroides se realiza aleatoriamente en una primera etapa de inicialización, y una vez clasificado todo el conjunto Y el clasificador ha de volver a ejecutarse para analizar si la elección de centroides que se hizo en la iteración anterior fue la correcta.

Una forma de evitar esta recurrencia es calcular la distancia entre todos los datos a clasificar (\vec{y}_i y \vec{g}_j son en ese caso dos datos del set, \vec{y}_1 e \vec{y}_2), de modo que se alarga el algoritmo de cálculo de distancias a cambio de eliminar la recursividad del proceso, y en consecuencia, la posibilidad de que éste no converja. Los segmentadores jerárquicos y los basados en la optimización de una función de coste, como el “*subtractive*” funcionan de esta segunda forma.

La Figura II.5 muestra una representación de la zona de distancia constante en un espacio bidimensional ($\lambda_T = 2$), como el elegido para la aplicación de interés, definido por la variable distancia a través de la ecuación de la norma d_δ , y en función de δ .

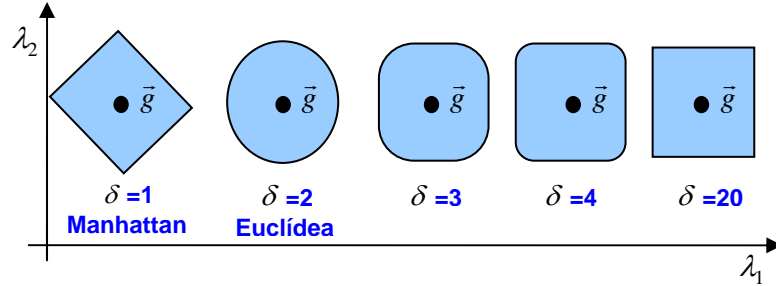


Figura II.5. Representación del entorno de distancia constante al centroide (punto negro \vec{g}) en un espacio bidimensional ($\lambda_T = 2$), definido por la ecuación <II.2> en función de δ .

Los métodos de medida de distancia más usados en aplicaciones de clasificación son las que se enumeran a continuación:

- **Distancia Manhattan** $d_{1|i,j} = d_{manhat|i,j}$: Esta forma de la variable distancia es la más fácil de calcular, y se obtiene sumando la diferencia de posición entre \vec{y}_i y \vec{g}_j en cada dimensión del espacio definido (<II.3>). Suele utilizarse cuando el espacio de definición de la distancia tiene muchas dimensiones, pues con otra forma de esta variable, el cálculo sería difícilmente ejecutable en tiempo real. En algunos casos se emplea una modificación de la medida de Manhattan básica $d_{1mod|i,j}$ como la que se muestra en la ecuación <II.4>, en la que el sumatorio de $d_{manhat|i,j}$ se calcula sobre el valor absoluto de la diferencia de posición entre \vec{y}_i y \vec{g}_j , de modo que no exista compensación entre las distancias en diferentes dimensiones del espacio definido; y se incluye un factor de normalización por dimensión (η_λ) para hacer la medida invariante a las diferencias de escala en las distintas dimensiones. Esta normalización es una técnica habitual en el cálculo de cualquier medida de distancia.

$$d_{manhat|i,j} = d_{1|i,j} = \sum_{\lambda=\lambda_1}^{\lambda_T} (a_{\lambda,i} - b_{\lambda,j}) \tag{II.3}$$

$$d_{1mod|i,j} = \sum_{\lambda=\lambda_1}^{\lambda_T} \eta_\lambda |a_{\lambda,i} - b_{\lambda,j}| \tag{II.4}$$

- **Distancia euclídea** $d_{2|i,j} = d_{euclidea|i,j}$: Se trata, probablemente, de la medida de distancia más utilizada en tareas de clasificación. Se define como la distancia geométrica entre dos puntos en un espacio multidimensional (<II.5>), \vec{y}_i y \vec{g}_j en este caso. La normalización previa de las variables que definen el espacio de medición de distancia es también una técnica habitual en el cálculo de la distancia euclídea.

$$d_{euclidea|i,j} = d_{2|i,j} = \sqrt{\sum_{\lambda=\lambda_1}^{\lambda_T} (a_{\lambda,i} - b_{\lambda,j})^2} \quad <II.5>$$

- *Distancia de Mahalanobis* $d_{Mahalanobis|i,j}$: Introducida por P. C. Mahalanobis en 1936, esta medida de distancia es una extensión de la euclídea ($d_{2|i,j} = d_{euclidea|i,j}$) al caso en que las variables referidas en las distintas dimensiones del espacio de comparación sean probabilísticas, de varianza σ_λ^2 en cada dimensión y matriz de covarianza $\Omega_{i,j}$ de las variables \bar{y}_i y \bar{g}_j . La ecuación <II.6> muestra la definición general de la distancia de Mahalanobis entre dos puntos, mientras que la expresión <II.7> es la simplificación de la anterior para el caso en que las variables implicadas estén incorreladas, situación en la que la matriz de covarianza es diagonal.

$$d_{Mahalanobis|i,j} = \sqrt{(\bar{y}_i - \bar{g}_j)' \Omega_{i,j}^{-1} (\bar{y}_i - \bar{g}_j)} \quad <II.6>$$

$$d_{Mahalanobis|i,j} = \sqrt{\sum_{\lambda=\lambda_1}^{\lambda_T} \frac{(a_{\lambda,i} - b_{\lambda,j})^2}{\sigma_\lambda^2}} \quad <II.7>$$

A la hora de elegir el tipo de distancia a usar en la clasificación es muy importante que ésta sea invariante a las rotaciones, las traslaciones y los escalados, de modo que pueda utilizarse en iguales condiciones para comparar todos los elementos del set de datos a organizar. De las tres medidas de distancia presentadas, solo la de Mahalanobis ($d_{Mahalanobis|i,j}$) cumple todas estas especificaciones, si bien la versión normalizada de la euclídea ($d_{euclidea|i,j}$) cumple con las de invarianza a la traslación y al cambio de escala.

La característica más importante que diferencia a la distancia de Mahalanobis ($d_{Mahalanobis|i,j}$) del resto es que incluye información sobre el ruido asociado y la correlación existente entre los datos a organizar. El inconveniente unido al uso de esta característica es el incremento de carga computacional que conlleva el cálculo de la matriz de covarianza $\Omega_{i,j}$ entre cada par de datos.

La inclusión de la matriz de covarianza $\Omega_{i,j}$ en el cálculo de la medida de distancia $d_{Mahalanobis|i,j}$ permite tener en cuenta la fiabilidad de los datos del set Y a clasificar a la hora de usarla en comparaciones como las que se realizan en el clustering. Este hecho resulta de especial interés cuando los datos implicados en la medida proceden de diferentes fuentes, pues en ese caso es necesario ponderar la distancia entre cada par de datos por la varianza del ruido asociado a cada una de las fuentes, a través de la matriz de covarianza $\Omega_{i,j}$. Si los datos del set Y están incorrelados y la varianza del ruido asociado a las fuentes de las que éstos se extraen es la misma, la distancia de Mahalanobis $d_{Mahalanobis|i,j}$ coincide con la euclídea $d_{2|i,j} = d_{euclidea|i,j}$.

En el capítulo 1 se establece como especificación de desarrollo de la tesis la posibilidad de utilizar distintos tipos de sensores para obtener la información de posición necesaria para el

seguimiento. Sin embargo, finalmente se especifica que la aplicación de MTT de interés se desarrolla únicamente a partir de datos de visión, por lo que el uso de la distancia de Mahalanobis ($d_{Mahalanobis|i,j}$) no es determinante.

Ante esta situación, se opta por usar la distancia euclídea ($d_{2|i,j} = d_{euclidea|i,j}$) como característica diferencial en la segmentación. Se plantea además como trabajo futuro, la aplicación de los algoritmos desarrollados a un espacio multisensorial, y el uso de la distancia de Mahalanobis $d_{Mahalanobis|i,j}$ en el proceso de clasificación, en ese caso.

II.4. Descripción de las variables utilizadas para la validación de los distintos algoritmos de clasificación

En general y de forma cualitativa, la segmentación de un conjunto Y será más efectiva cuanto más parecidos sean los miembros de cada clase j (L_j) entre ellos y más diferentes sean a los del resto de clases generadas como salida del algoritmo, de acuerdo con la característica diferencial definida.

La implicación directa de la definición anterior, ya comentada en el capítulo 5, es que la eficacia de cualquier clasificador depende directamente de la característica de distancia elegida para realizar la segmentación. Por otro lado, la efectividad de la clasificación dependerá también del algoritmo utilizando y, en algunos casos, de su inicialización (caso de algoritmos no repetibles o de número fijo de grupos).

Desde un punto de vista cuantitativo, los parámetros utilizados más habitualmente para la comparación de diferentes métodos de clasificación son los que se muestran a continuación ([Everitt01], [Wilson90]):

- *Número de clases de salida k* : Es un parámetro comparativo únicamente válido en caso de que se conozca el número de clases en que han de quedar organizados los datos del set Y , por lo que suele usarse solo en aplicaciones reales en las que se cuente con un resultado de referencia o “*ground truth*”. En la aplicación de MTT de interés en la tesis, se puede utilizar este parámetro pues el “*ground truth*” es conocido (se obtiene de la asignación manual del conjunto Y a los diferentes objetos en el entorno de los que es extraído).
- *Densidad o Dispersión de la clase*: Se trata de un parámetro cuya definición depende de la referencia literaria consultada, pero que suele relacionarse en general con el número de miembros asignados a cada clase j (l_j). Lo más habitual es que se calcule como el sumatorio de la medida de distancia asociada a cada uno de los elementos que conforman cada cluster j ($d_{i,j}$), como se muestra en <II.8>. Esta medida puede estar referida a la distancia entre el centroide del cluster (\bar{g}_j) y cada uno de sus miembros (L_j), o entre los distintos miembros del cluster j , como se expone en el capítulo 5. Un valor pequeño de este parámetro indica, en general, una mayor eficiencia en la segmentación. Además, sumando

los valores de dispersión de todos los grupos generadas en la clasificación, se obtiene la dispersión global de ese proceso (<II.9>), también conocida como “*distorsión*” o “*error cuadrático medio*” de la clasificación. Finalmente, con objeto de normalizar el parámetro de dispersión y facilitar así su uso en la comparación de distintos segmentadores, se suele utilizar el valor de distorsión para redefinir el cálculo de la dispersión como se muestra en la ecuación <II.10>.

$$dispersion_j = \sum_{i=1}^{l_j} d_{i,j} \quad \langle \text{II.8} \rangle$$

$$distorsion = \sum_{j=1}^k dispersion_j = \sum_{j=1}^k \sum_{i=1}^{l_j} d_{i,j} \quad \langle \text{II.9} \rangle$$

$$dispersion_j = \frac{\sum_{i=1}^{l_j} d_{i,j}}{distorsion} \quad \langle \text{II.10} \rangle$$

El algoritmo “*k-medias*” permite obtener la organización óptima del set Y , debido a que su funcionalidad se basa en la búsqueda del valor mínimo del parámetro de dispersión.

- *Separación entre clases*: Este parámetro se refiere más a la relación que aparece al final de la clasificación entre los distintos grupos formadas, que a los propios grupos en sí. Existen, también en este caso, distintas definiciones para este parámetro en la literatura científica, si bien la más extendida es semejante a la mostrada para el parámetro de dispersión, sólo que en este caso se suman las distancias entre los centroides ($\vec{g}_j / j = 1:k$) de los distintas clusters (<II.11>). En [Moore01] se expone otra formulación para este parámetro, que tiene en cuenta la dispersión de cada cluster (<II.12>). Un valor grande del parámetro será indicativo de una correcta clasificación.

$$separacion = \sum_{k1=1}^k \sum_{k2=1}^k d(\vec{g}_{k1}, \vec{g}_{k2}) \quad \langle \text{II.11} \rangle$$

$$separacion = \sum_{k1=1}^k \frac{\sum_{k2=1}^k d(\vec{g}_{k1}, \vec{g}_{k2})}{dispersion_{k1}} \quad \langle \text{II.12} \rangle$$

Al margen de los parámetros cuantitativos presentados, una característica cualitativa muy valorada en los clasificadores es que permitan separar los datos del set Y que solo contienen ruido (outliers) y asociarlos a un grupo específico o eliminarlos en el propio segmentador.

Esta característica es generalmente propia de los segmentadores borrosos, como el “*subtractive*”, en los que los outliers se caracterizan por tener un valor de pertenencia nulo o muy bajo para todos los clusters. En algoritmos booleanos como “*k-medias*” esta cualidad es mucho más difícil de conseguir, como se analiza en el capítulo 5.

APÉNDICE III

DESCRIPCIÓN DE LOS EXPERIMENTOS

En esta parte del documento se describen los experimentos usados en los capítulos 5, 6 y 7 para valorar la bondad de los algoritmos propuestos en las distintas etapas del XPFCP analizadas, y de otros usados en la comparación empírica de este seguidor multi-objeto.

El objetivo de la definición de varias situaciones de testeo específicas es usar una misma casuística para comparar el comportamiento de las distintas alternativas propuestas. Con este fin, los experimentos elegidos abarcan, además, situaciones de distinta complejidad.

En este apéndice se incluye además un listado de los distintos tipos de errores que aparecen en los experimentos de MTT descritos. Se define además la importancia relativa de cada uno de ellos en la aplicación de seguimiento de interés.

El proceso de tracking ha sido realizado en todos los experimentos sobre un procesador Intel Centrino DUO a 1.8GHz. Además, los dos sistemas de observación usados (ver capítulo 1) se basan en la captura de imágenes a una velocidad de 15FPS, en blanco y negro de 320x240 píxeles para el caso del sistema centralizado, y en color y de 640x480 píxeles en el caso del distribuido. Los robots usados son plataformas comerciales (Pioneer de la empresa MobileRobots¹) con control autónomo de movimiento y, en las pruebas con observación

¹ www.mobilerobots.com

centralizada, el robot que se mueve por el entorno lleva incorporadas dos cámaras a una altura de 1.5m del suelo y con una separación entre ellas de 30cm. En [Marrón06a], [Marrón07a] y [Pizarro08] se describen en mayor detalle las plataformas de observación usadas para realizar estos experimentos.

III.1. Experimentos generales usados

Para que los resultados obtenidos con los experimentos aquí expuestos tengan validez y generalidad a la hora de obtener conclusiones a partir de ellos, es necesario que incluyan situaciones reales y variadas. Esta razón lleva a definir los cinco experimentos generales que se describen a continuación.

Los cuatro primeros de éstos se obtienen a partir de un sistema de observación centralizada (ver en el capítulo 1) cuya funcionalidad se explica en [Marron06a] o [Marrón07a]. En los párrafos que siguen se describen cada uno de estos tests, y en la Tabla III.1 se resumen las características más interesantes de todos ellos:

Tabla III.1. Resumen de características de los experimentos usados.

Experimento	Nº frames (tiempo)	Nº objetos	Tasa de oclusiones parciales/totales	Tasa de error de sensado	Tasa de outliers (tasa y tiempo de los no espurios)
Completo corto	302 (20.13s)	de 2 a 5 (4 móviles 1 estático)	45% 1 o 2 objetos, 10%	25%	22% (10%)
Complejo	101 (6.73s)	de 3 a 4 (móviles)	87% 1 o 2 objetos, 11%	0%	0%
Completo largo	1098 (1min 13s)	de 0 a 5 (5 móviles 1 estático)	39% hasta 5 objetos, 10%	17%	11% (17% 200ms-266ms)
Ruido	72 (4.8s)	2 (móviles)	0%	0%	60% (39% 200ms-266ms)
Observación distribuida	1180 (1min 19s)	de 2 a 4 (móviles)	33% 2 a 4 objetos (solapamientos)	0%	0%

1. *Experimento completo corto*: Aparece un número de objetos variable entre 2 y 5 (hasta 4 personas, y 1 papelera). Inicialmente sólo se encuentra una papelera y una persona en la escena que se va acercando a la cámara y a la que se le unen al final del test otras 3. Este experimento incluye situaciones de complejidad variable, que incluyen oclusiones, situaciones con medidas repartidas entre los objetos de forma desigual que producen la desaparición de todas las medidas asociadas a algún/os objeto/s sin razón aparente (tasa de error de sensado, en la Tabla III.1) y outliers debidos a la baja fiabilidad del sistema de adquisición. Este experimento ha sido elegido por la variedad de las situaciones que presenta, dentro de su corta duración, que ejemplifican adecuadamente las típicas de un problema de tracking en entornos complejos e interiores.
2. *Experimento complejo*: Aparecen inicialmente, en iteraciones consecutivas y a poca distancia del sistema de observación, 3 personas que se van juntando (durante el 32% de las iteraciones del test) y a las que después se añade una 4ª, manteniéndose las 4 durante el

resto del experimento. En este experimento no aparecen outliers, pero la desigual distribución de medidas entre los objetos tiene una repercusión aún mayor en el funcionamiento del algoritmo de MTT debido a las continuas situaciones de oclusión. Este test se ha elegido para probar el funcionamiento de los algoritmos en una situación de continua complejidad de tracking. Este tipo de situaciones aparece con bastante frecuencia en aplicaciones robóticas como las de interés en la tesis, si bien su presencia no es constante en estas aplicaciones. Debido a ello, los experimentos completo corto y largo constituyen una representación más general de las situaciones típicas de tracking de interés.

3. *Experimento completo largo*: Este test incluye a los dos anteriores, por lo que presenta situaciones de mayor y menor complejidad, con oclusiones parciales y totales, outliers y escenas en las que desaparecen todas las medidas asociadas algún/os objeto/s debido a errores de sensado. Este experimento ejemplifica, por tanto, un problema típico de tracking en entornos interiores muy poblados como es de interés en la tesis, por lo que los resultados obtenidos al ejecutar los algoritmos bajo prueba en él son una muestra bastante general de su comportamiento en cualquier otra situación.
4. *Experimento de Ruido*: Este test presenta una alta tasa de outliers, que además permanecen durante un tiempo relativamente alto. Por lo demás es un experimento sencillo de tracking de 2 personas sin oclusiones. El test se ha elegido para probar específicamente el comportamiento de los algoritmos frente a la aparición de outliers o ruidos, pero no representa una situación normal de tracking en el entorno de aplicación de interés en la tesis.

De las descripciones anteriores se desprende que, en general, los experimentos usados para poner a prueba los distintos algoritmos incluyen situaciones algo más complejas que las normalmente presentes en situaciones reales de tracking, pero que permiten asegurar la validez de los resultados obtenidos en cualquier tipo de situación.

Debido a ello, y con objeto de obtener resultados comparables con los expuestos en otras publicaciones del área de interés, en los datos finales de fiabilidad obtenidos para el XPFCP en el capítulo 6 se obtienen tasas de fiabilidad de éste en situaciones continuas de error. De este modo se computan solamente los errores que genera el algoritmo de MTT de forma sostenida durante un tiempo suficientemente largo como para que el error pueda generar problemas en los procesos robóticos que usen como entrada la salida del seguidor.

En el capítulo 6 se utiliza otra prueba para validar el comportamiento del XPFCP con un sistema de observación diferente al usado en los tests antes descritos (en [Pizarro08] se detalla su funcionamiento). Las características más importantes de este test se listan en la Tabla III.1, presentándose a continuación una descripción general del experimento:

5. *Experimento con observación distribuida*: En este experimento aparece un número variable de entre 2 y 4 móviles entre los que se encuentran 3 personas y un robot. Gracias al sistema de observación utilizado no aparecen outliers en el set de datos Y_t en este caso, pues éstos son eliminados al combinar la información de las cuatro cámaras que conforman el sistema de sensado. Además tampoco se producen situaciones de oclusión, pero sí continuos solapamientos de los datos de posición que producen la duplicación de las hipótesis asociadas a los objetos de los que se obtienen estos datos (ver en el capítulo 6).

Las situaciones de tracking que aparecen en este último test incluyen, como se observa, dificultades distintas a las presentes en los tests realizados con el sistema de observación centralizado. Esta última prueba completa, por tanto, la diversidad de situaciones de tracking más habituales, incluso con distintos sistemas de observación, en la aplicación del XPFCP a la tarea de interés, de modo que los resultados obtenidos con el seguidor propuesto en la tesis, en el conjunto de las cinco pruebas descritas, es extrapolable a cualquier situación de MTT.

III.2. Tipos de errores generalizados

En el análisis de los resultados obtenidos al ejecutar los distintos algoritmos de tracking y clasificación en las pruebas descritas en el punto anterior, se mencionan 4 distintos tipos de errores, cuyo significado se detalla en el listado siguiente, en el que éstos aparecen por orden creciente de importancia en la aplicación de interés:

1. *Duplicación*: Indica que se asignan dos clases o hipótesis de seguimiento a un solo objeto detectado en la escena.
2. *Desplazamiento*: Indica que el centroide de la clase generada está desplazado, en el espacio correspondiente, respecto al calculado manualmente de las medidas obtenidas de él por el sistema de observación, al menos una distancia igual a la de gating (ver capítulo 2).
3. *Unión de 2 en 1*: Indica que, generalmente debido a un error de desplazamiento o una oclusión, se crea una sola clase para 2 objetos detectados en la escena.
4. *No generado*: Indica que no se genera clase o hipótesis de seguimiento para un objeto (detectado por el sistema de observación usado), en general, debido a problemas de oclusión o pobre sensado de éste.

Es importante tener en cuenta la relevancia relativa de estos cuatro tipos de errores. Así, el error de duplicación es el menos relevante, pues puede eliminarse mediante un post-proceso ([Vremaak03]). El error de desplazamiento le sigue en importancia pues, en la aplicación robótica en la que se enmarca esta tesis, es fundamental conocer con la mayor exactitud posible la ubicación de los objetos para alcanzar la seguridad requerida. Finalmente, los errores de unión de 2 clases en 1 o de no generación de clases o hipótesis de seguimiento son los más importantes, pues indican un fallo de tracking.

Por otro lado, los resultados generados por el PF en su aplicación multimodal a la tarea de seguimiento (comentados en varios análisis en los apartados 6.1.3 y 6.1.4 del capítulo 6) sólo aparecen errores del último tipo, pues son los únicos que tienen sentido al analizar la solución probabilística de tracking representada por el set de partículas S_t .

Es también importante destacar que los algoritmos probados filtran el conjunto de medidas Y_t , por lo que, en general, retardan la salida correcta generando la mayor parte de estos errores ([Schulz03b]). Debido a ello, como se comenta en el apartado anterior, es habitual indicar la fiabilidad de los algoritmos teniendo en cuenta solamente aquellos errores que se producen de forma sostenida durante un tiempo concreto.

BIBLIOGRAFÍA

- [Almeida05] A. Almeida, J. Almeida, R. Araujo. "Real-time tracking of moving objects using particle filters", Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE05), ISBN: 0-7803-8738-4, Vol. 4, pp. 1327-1332, Dubrovnik, June 2005.
- [Alsabti98] K. Alsabti, S. Ranka, V. Singh. "An efficient k-means clustering algorithm", Proceedings of the First Workshop on High Performance Data Mining at the International Parallel Processing Symposium (IPPS98), Orlando, 1998.
- [Arkin98] R.C. Arkin. "Behaviour-based robotics", The MIT Press, ISBN: 0262011654, Cambridge, 1998.
- [Arulampalam02] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp. "A tutorial on particle filters for online nonlinear non-gaussian bayesian tracking", IEEE Transactions on Signal Processing, Vol. 50, nº 2, pp. 174-188, February 2002.
- [Bar-Shalom88] Y. Bar-Shalom, T. E. Fortmann. "Tracking and data association", Mathematics in Science and Engineering, Vol. 179, Academic Press, ISBN 0-12-079760-7, 1988.
- [Bar-Shalom90] Y. Bar-Shalom. "Multitarget-Multisensor tracking: advances applications", Vol. 1, Artech House, ISBN: 089006377X, Norwood 1990.

- [Bar-Shalom93] Y. Bar Shalom, X. R. Li. "Estimation and tracking principles techniques and software", Artech House, ISBN 0-89006-643-4, Boston, 1993.
- [Berkhin02] P. Berkhin. "Survey of clustering data mining techniques", Technical Report, Accrue Software, 2002.
- [Bezdek81] J.C. Bezdek. "Pattern recognition with fuzzy objective function algorithms", Plenum Press, ISBN: 0306406713, New York, 1981.
- [Blackman99] S. Blackman, R. Popoli. "Design and analysis of modern tracking systems", Artech House, ISBN: 1580530060, Norwood, 1999.
- [Bolic03] M. Bolic, P.M. Djuric, S. Hong. "New resampling algorithms for particle filters", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP03), Vol. 2, pp. 589-592, ISBN: 0-7803-7663-3, Hong Kong, April 2003.
- [Branson05] K. Branson, S. Belongie. "Tracking multiple mouse contours (without too many samples)", Proceedings of the Fourth IEEE Conference on Computer Vision and Pattern Recognition (CVPR05), ISBN: 0-7695-2372-2, Vol. 1, pp. 1039-1046, San Diego, June 2005.
- [Bravo07] I. Bravo. "Arquitectura basada en FPGAs para la detección de objetos en movimiento, utilizando visión computacional y técnicas PCA", Tesis Doctoral, Universidad de Alcalá, 2007.
- [Burgard99] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyery, D. Schulz, W. Steiner, S. Thrun. "Experiences with an Interactive Museum Tour Guide", Journal on Artificial Intelligence, Vol. 114, nº 1, pp. 3-55, 1999.
- [CARE-O-BOT] Care-o-bot Project. <http://www.care-o-bot.de/english/>
- [Carpenter99] J. Carpenter, P. Clifford, P. Fearnhead. "An improved particle filter for non-linear problems", IEE Proceedings on Radar, Sonar and Navigation, Vol. 146, pp. 2-7, 1999.
- [Castellanos99] J.A. Castellanos, J.D. Tardós. "Mobile robot localization and map building: a multisensor fusion approach", Kluwer Academic Publishers, ISBN 0-7923-7789-3, Boston, 1999.
- [Cepolina03] F. Cepolina, R.C. Michelini, R.M. Molfino, R.P. Razzoli. "Gecko, a climbing robot for walls cleaning", Proceedings of 1st International Workshop on Advances in Service Robotics (ASER03), ISBN 3-8167-62689, Bardolino, March 2003.

- [Chang06] Y. Chang, C. Hu, R. Feris, M. Turk. "Manifold based analysis of facial expression", *Journal of Image and Video Computing*, Vol. 24, n° 6, pp. 605-614, June 2006.
- [Chen02] Y. Chen, T.S. Huang, Y. Rui. "Mode-based multi-hypothesis head tracking using parametric contours", *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR02)*, ISBN: 0-7695-1602-5, Washington, May 2002.
- [Chiu94] S.L. Chiu. "Fuzzy model identification based on cluster estimation", *Journal of Intelligent & Fuzzy Systems*, Vol. 2, n° 3, pp. 267-278, September 1994.
- [Chiu96] S.L. Chiu. "Selecting input variables for fuzzy models", *Journal of Intelligent & Fuzzy Systems*, Vol. 4, n° 4, September 1996.
- [Coraluppi04] S. Coraluppi, C. Carthel. "Multi-hypothesis sonar tracking", *Proceedings of the Seventh International Conference on Information Fusion (FUSION04)*, ISBN: 91-7056-115-X, Vol. 1, pp. 33-40, Stockholm, June 2004.
- [Cox96] I. Cox, S. Hingorani. "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, n° 2, pp. 138-150, February 1996.
- [DA VINCI] DA VINCI Surgical System. <http://www.intuitivesurgical.com/>
- [Dellaert03] F. Dellaert. "A sample of Monte-Carlo methods in robotics and vision", *Proceedings of the International Symposium on the Science of Modelling (ISM03)*, 2003.
- [Deriche90] R. Deriche and O. Faugeras, "Tracking line segments", *Proceedings of the First European Conference on Image and Vision Computing*, ISBN: 0-387-52522-X, Vol. 8, n° 4, pp. 261-270, Antibes, 1990.
- [Douc05] R. Douc, O. Cappé, E. Moulines. "Comparison of resampling schemes for particle filtering", *Proceedings of the Fourth International Symposium on Image and and Signal Processing and Analysis (ISPA05)*, pp. 64-69, Zagreb, September 2005.
- [Doucet97] A. Doucet. "Monte-Carlo methods for bayesian estimation of Hidden-Markov models. Application to Radiation Signal", PhD Thesis, University of Paris-Sud, France, 1997.
- [Doucet00] A. Doucet, S. Godsill, C. Andrieu. "On sequential Monte-Carlo sampling methods for bayesian filtering", *Statistics and Computing*, Vol. 10, n° 3, pp. 197-208, 2000.

- [Doucet01] A. Doucet, N. de Freitas, N. Gordon. "Sequential Monte-Carlo methods in practice", Springer-Verlag, ISBN: 0-387-95146-6, New York, 2001.
- [Dubois97] D. Dubois, H. Prade, R.R. Yager. "Fuzzy information engineering: a guided tour of applications", John Wiley & Sons, ISBN: 0471147664, New York, 1997.
- [Dunn74] J.C. Dunn. "A fuzzy relative of the ISODATA process and its use in detecting compact well separated clusters", Journal of Cybernetics, Vol. 3, nº 3, pp. 32-57, 1974.
- [Durrant-Whyte06] H.F. Durrant-Whyte, T. Bailey. "Simultaneous localization and mapping", IEEE Robotics and Automation Magazine, Vol. 13, nº 2, pp. 99-108, June 2003.
- [Elfes87] A. Elfes. "Sonar-based real-world mapping and navigation", IEEE Journal of Robotics and Automation, Vol. 3, nº 3, pp: 249-265, June 1987.
- [Everitt01] B. Everitt, S. Landau, M. Leese. "Cluster analysis", Edward Arnold Publishers, 4th Edition, ISBN: 0-340-76119-9, London, 2001.
- [Fernández08] D. Fernández. "Sistema de detección de peatones mediante visión estereoscópica para la asistencia a la conducción", Tesis Doctoral, Universidad de Alcalá, 2008.
- [Fleuret08] F. Fleuret, J. Berclaz, R. Lengagne, P. Fua. "Multi-camera people tracking with a probabilistic occupancy map", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 30, nº2, pp. 267-282, February 2008.
- [Fortmann83] T. Fortmann, Y. Bar-Shalom, M. Scheffe. "Sonar tracking of multiple targets using joint probabilistic data association", IEEE Journal of Oceanic Engineering, Vol. 8, pp. 173-184, July 1983.
- [Fox99a] D. Fox, W. Burgard, S. Thrun. "Markov localization for mobile robots in dynamic environments", Journal on Artificial Intelligence Research, Vol. 11, pp. 391-427, 1999.
- [Fox99b] D. Fox, W. Burgard, F. Dellaert, S. Thrun. "Monte-Carlo localization. Efficient position estimation for mobile robots" Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI99), Orlando, July 1999.
- [Fox03] D. Fox, J. Hightower, L. Liao, D. Schulz, G. Borriello. "Bayesian filtering for location estimation", IEEE Pervasive Computing, Vol. 2, nº 3, pp. 24-33, IEEE Computer Society Press, June 2003.

- [Frank03] O. Frank, J. Nieto, J. Guivant, S. Scheduling. "Multiple target tracking using sequential Monte-Carlo methods and statistical data association", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS03), ISBN: 0-7803-7860-1, Vol. 3, pp. 2718- 2723, Las Vegas, October 2003.
- [Fujita98] M. Fujita, H. Kitano. "Development of an autonomous quadruped robot for robot entertainment", Autonomous Robots, Vol. 5, n° 1, pp.7-18, March 1998.
- [García01] J.C. García. "Sistema de posicionamiento y auto-localización para sillas de ruedas autónomas", Tesis Doctoral, Universidad de Alcalá, 2001.
- [Gilks96] W.R. Gilks, S. Richardson, D.J. Spiegelhalter. "Markov chain Monte-Carlo in practice", Chapman and Hall, ISBN: 0412-05551-1, London, 1996.
- [Gordon93] N.J. Gordon, D.J. Salmond, A.F.M. Smith. "Novel approach to nonlinear/non-gaussian bayesian state estimation", IEE Proceedings-F in Radar and Signal Processing, Vol. 140, n° 2, pp. 107-113, April 1993.
- [Gordon97] N.J. Gordon. "A hybrid bootstrap filter for target tracking in clutter", IEEE Transactions on Aerospace and Electronic Systems, Vol. 33, n° 1, pp. 353-358, January 1997.
- [Graf04] B. Graf, M. Hans, R.D. Schraft. "Care-O-bot II - Development of a next generation robotic home assistant", Autonomous Robot Journal, Vol. 16, n° 2, pp. 193-205, March 2004.
- [Grisetti05] G. Grisetti, C. Stachniss, W. Burgard. "Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling", Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA05), ISBN: 0-7803-8914-X, Vol. 5, pp: 2432-2437, Barcelona, April 2005.
- [Gustafsson02] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, P.J. Nordlund. "Particle Filters for Positioning, Navigation and Tracking", IEEE Transactions on Signal Processing, Vol. 50, n° 2, pp.425-437, February 2002.
- [Hammersley54] J.M. Hammersley, K.W. Morton. "Poor's man Monte-Carlo", Journal of the Royal Statistical Society. Vol. 16, pp. 23-38. 1954.
- [Handschin69] J.E. Handschin, D.Q. Mayne. "Monte-Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering", International Journal of Control, Vol. 9, n° 5, pp. 547-559, 1969.

- [Hertz91] J. Hertz, A. Krogh, R.G. Palmer. "Introduction to the theory of neural computation", Addison Wesley, ISBN: 1-201-51560-1, 1991.
- [Hirose00] S. Hirose, H. Ohno, T. Mitsui, K. Suyama. "Design of in-pipe inspection vehicles for $\varnothing 25$, $\varnothing 50$, $\varnothing 150$ pipes", Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA99), ISBN: 0-7803-5180-0, Vol. 3, pp. 2309-2314, Detroit, May 1999.
- [Hol06] J.D. Hol, T.B. Schön, F. Gustafsson. "On resampling algorithms for particle filters", Proceedings of the Nonlinear Statistical Signal Processing Workshop (NSSPW06), Cambridge, September 2006.
- [Holland75] J.H. Holland. "Adaptation in natural and artificial systems", University of Michigan Press, Ann Arbor, 1975.
- [Hue01] C. Hue, J.P. Le Cadre, P. Pérez. "A particle filter to track multiple objects", Proceedings of the IEEE Workshop on Multi-Object Tracking, ISBN: 0-7695-1171-6, pp. 61-68, Vancouver, August 2001.
- [Hue02] C. Hue, J.P. le Cadre, P. Pérez. "Tracking multiple objects with particle filtering", IEEE Transactions on Aerospace and Electronics Systems, Vol. 38, n° 3, pp. 791-812, July 2002.
- [Illmann01] J. Illmann, B. Kluge, E. Prassler. "People tracking using omni-directional vision and range data", Proceedings of the SPIE Fourth International Conference on Sensor Fusion and Decentralized Control in Robotic Systems, Vol. 4571, pp. 128-138, Newton, November 2001.
- [Isard96] M. Isard, A. Blake. "Contour tracking by stochastic propagation", Proceedings of the Fourth European Conference on Computer Vision (ECCV96), pp. 343-356, Cambridge, October 1996.
- [Isard98a] M. Isard, A. Blake. "Condensation: Conditional density propagation for visual tracking", International Journal of Computer Vision, Vol. 29, n° 1, pp. 5-28, 1998.
- [Isard98b] M. Isard, A. Blake. "ICondensation: Unifying low-level and high-level tracking in a stochastic framework", Proceedings of the Fifth European Conference on Computer Vision (ECCV98), Vol. 1, pp. 893-908, 1998.
- [Isard01] M. Isard, J. MacCormick. "BraMBLe: A bayesian multiple-blob tracker", Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV01), Vol. 2, pp. 34-41, Vancouver, July 2001.
- [Ishida03] T. Ishida. "A small biped entertainment robot SDR-4X II", Proceedings of the 2003 IEEE International Symposium on Computational Intelligence in

- Robotics and Automation (CIRA03), ISBN: 0-7803-7866-0, Vol. 3, pp. 1046-1051, Kobe, July 2003.
- [Julier97] S.J. Julier and J.K. Uhlmann. "A new extension of the Kalman filter to nonlinear systems", Proceedings of the SPIE Eleventh International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multisensor Fusion, Tracking and Resource Management (AeroSense97), Vol. 2, Orlando, April 1997.
- [Jain99] A.K. Jain, M.N. Murty, P.J. Flynn. "Data clustering: a review", ACM Computing Surveys, Vol. 31, n° 3, pp 264-323, September 1999.
- [Kainulainen02] J. Kainulainen. "Clustering algorithms: basics and visualization", Notes of the Laboratory of Computer and Information Science, Helsinki, University of Technology, August 2002.
- [Kalman60] R.E. Kalman. "A new approach to linear filtering and prediction problems", Transactions of the ASME Journal of Basic Engineering, Vol. 82, n° D, pp. 35-45, March 1960.
- [Kanungo02] T. Kanungo, N.S. Netanyahu, A.Y. Wu. "An efficient k-means clustering algorithm: analysis and implementation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, n° 7, July 2002.
- [Karlsson01] R. Karlsson, F. Gustafsson. "Monte-Carlo data association for multiple target tracking", Proceedings of IEE Target Tracking: Algorithms and applications (IEESEM01), Vol. 1, pp. 13/1-13/5, Amsterdam, October, 2001.
- [Kaufman90] L. Kaufman, P.J. Rousseeuw. "Finding groups in data: An introduction to cluster analysis", John Wiley & Sons, ISBN: 0471878766, New York, 1990.
- [Khan03] Z. Khan, T. Balch, F. Dellaert. "Efficient particle filter-based tracking of multiple interacting", Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS03), ISBN: 0-7803-7860-1, Vol. 1, pp. 254-259, Las Vegas, October 2003.
- [Khan04] Z. Khan, T. Balch, F. Dellaert. "A Rao-Blackwellized particle filter for Eigen tracking", Proceedings of the Third IEEE Conference on Computer Vision and Pattern Recognition (CVPR04), ISBN: 0-7695-2158-4, Vol. 1, pp. 980-986, Washington, June 2004.
- [Khan05] Z. Khan, T. Balch, F. Dellaert. "MCMC-Based particle filtering for tracking a variable number of interacting targets", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, n° 11, pp. 1805-1918, November 2005.

- [Khan06] S.M. Khan, M. Shah. "A multiview approach to tracking people in crowded scenes using a planar homography constraint", Proceedings of the Ninth European Conference of Computer Vision (ECCV06), Lecture Notes in Computer Science, ISBN: 3-540-33838-1, Vol. 3954, Part IV, pp. 133-146, Graz, May 2006.
- [Kitagawa96] G. Kitagawa. "Monte-Carlo filter and smoother for non-Gaussian nonlinear state space models", Journal of Computational and Graphical Statistics, Vol. 5, pp. 1-5, 1996.
- [Kobayashi06] Y. Kobayashi, D. Sugimura, Y. Sato, K. Hirasawa, N. Suzuki, H. Kage, A. Sugimoto, "3D head tracking using the particle filter with cascaded classifiers", Proceedings of the British Machine Vision Conference (BMVC06), ISBN: 1-904410-14-6, Vol. 1, pp. 37-46, Edinburgh, September 2006.
- [Kohonen82] T. Kohonen. "Clustering, taxonomy, and topological maps of patterns", Proceedings of the Sixth International Conference on Pattern Recognition (ICPR82), pp. 114-128, Munich, 1982.
- [Koller-Meier01] E.B. Koller-Meier, F. Ade. "Tracking multiple objects using a condensation algorithm", Journal of Robotics and Autonomous Systems, Vol. 34, pp. 93-105, February 2001.
- [Kong94] A. Kong, J.S. Liu, R. Chen. "Sequential imputations and bayesian missing data problems", Journal of the American Statistical Association. Vol. 89, pp. 278-288, 1994.
- [Kotecha03] J.H. Kotecha, P.M. Djuric. "Gaussian particle filtering", IEEE Transactions on Signal Processing, Vol. 51, n° 10, pp. 2592-2601, October 2003.
- [Kreucher05] C. Kreucher, K. Kastella, A.O. III Hero. "Multitarget tracking using the joint multitarget probability density", IEEE Transactions on Aerospace and Electronic Systems, Vol. 41, n° 4, pp. 1396-1414, October 2005.
- [Lee99] J.H. Lee, N. Ando, H. Hashimoto. "Intelligent Space for Mobile Robots", Proceedings of the 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM99), ISBN: 0-7803-5040-5, pp.85-90, Atlanta, September 1999.
- [Leonard91] J.J. Leonard, H.F. Durrant-Whyte. "Simultaneous map building and localization for an autonomous mobile robot", Proceedings of the 1991 IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS91), ISBN: 0-7803-0067-X, Vol. 3, pp. 1442-1447, Osaka, November 1991.

- [LeSaux02] B. LeSaux and N. Boujemaa. "Unsupervised robust clustering for image database categorization", Proceedings of the Sixteenth International Conference on Pattern Recognition (ICPR02), ISBN: 0-7695-1695-X, Vol. 1, pp. 10259, Quebec, August 2002.
- [Liu98] J.S. Liu, R. Chen. "Sequential Monte-Carlo methods for dynamic systems", Journal of the American Statistical Association, Vol. 93, pp. 1032-1044, 1998.
- [Liu03] W.Y. Liu, C.J. Xiao, B.W. Wang, Y. Shi, S.F. Fang. "Study on combining subtractive clustering with fuzzy c-means", Proceedings of the Second International Conference on Machine Learning and Cybernetics, ISBN: 0-7803-7865-2, Vol. 5, pp. 2659-2662, November 2003.
- [López04] E. López. "Sistema de navegación global basado en procesos de decisión de markov parcialmente observables. Aplicación a un robot de asistencia personal", Tesis Doctoral, Universidad de Alcalá, 2004.
- [MacCormick99] J. MacCormick, A. Blake. "A probabilistic exclusion principle for tracking multiple objects", Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV99), Vol. 1, pp. 572-578, Corfu, September 1999.
- [MacCormick00] J. MacCormick, M. Isard. "Partitioned sampling, articulated objects, and interface-quality hand tracking", Proceedings of the Sixth European Conference on Computer Vision (ECCV00), Vol. 2, pp. 3-19, Dublin, June 2000.
- [MacQueen67] J.B. MacQueen. "Some methods for classification and analysis of multivariate observations", Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, pp. 281-297, Berkeley, 1967.
- [Maksarov95] D. Maksarov, H. Durrant-Whyte. "Mobile vehicle navigation in unknown environments: a multiple hypothesis approach", IEE Proceedings on Control Theory Application, Vol. 142, n° 4, pp. 385-400, July 1995.
- [Marrón04a] M. Marrón, M.A. Sotelo, J.C. García. "A probabilistic multimodal algorithm for tracking multiple and dynamic objects", in "Robotics: Trends, principles and applications", TSI Press, ISBN: 1-889335-21-5, pp. 511-516, Albuquerque, 2004.
- [Marrón04b] M. Marron, M.A. Sotelo, J.C. García. "Design and applications of an extended particle filter with a pre-clustering process, XPFCP", Proceedings of the IEEE Conference on Mechatronics & Robotics 2004 (MECHROB04), ISBN: 3-938153-50-X, Vol. 2/4, pp. 187-191, Aachen, September 2004.

- [Marrón04c] M. Marrón, M.A. Sotelo, J.C. García. "Tracking multiple and dynamic objects with an extended particle filter and an adapted k-means clustering algorithm", Preprints of the Fifth IFAC/EURON Symposium on Intelligent Autonomous Vehicles (IAV04), Lisboa, July 2004.
- [Marrón05a] M. Marrón, M.A. Sotelo, J.C. García, D. Fernández, D. Pizarro, "XPFCP: An extended particle filter for tracking multiple and dynamic objects in complex environments", Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS05), ISBN: 0-7803-9252-3, pp. 234-239, Edmonton, August 2005.
- [Marrón05b] M. Marrón, J.C. García, M.A. Sotelo, D. Fernández, D. Pizarro. "XPFCP: An extended particle filter for tracking multiple and dynamic objects in complex environments", Proceedings of the IEEE International Symposium on Industrial Electronics 2005 (ISIE05), ISBN: 0-7803-8738-4, Vol. I-IV, pp. 1587-1593, Dubrovnik, June 2005.
- [Marrón05c] M. Marrón, J.C. García, M.A. Sotelo, E.J. Bueno. "Clustering methods for 3D vision data and its application in a probabilistic estimator for tracking multiple objects", Proceedings of the Thirty First Annual Conference of the IEEE Industrial Electronics Society (IECON05), ISBN: 0-7803-9252-3, pp. 2017-2022, Raleigh, November 2005.
- [Marrón06a] M. Marrón, M.A. Sotelo, J.C. García, D. Fernández, I. Parra. "3D-Visual Detection of Multiple Objects and Environmental Structure in Complex and Dynamic Indoor Environments", Proceedings of the Thirty Second Annual Conference of the IEEE Industrial Electronics Society (IECON06), ISBN: 1-4244-0136-4, pp. 3373-3378, Paris, November 2006.
- [Marrón06b] M. Marrón, M.A. Sotelo, J.C. García. "Combined stochastic-deterministic solution for tracking multiple objects with an stereo-vision system", WSEAS Transactions on Signal Processing, Vol. 2, n°2, pp. 253-260, 2006.
- [Marrón06c] M. Marrón, M.A. Sotelo, J.C. García, J.L. Martín. "A bayesian solution to track multiple and dynamic objects robustly from visual data", Proceedings of the Third International IEEE Conference on Intelligent Systems (IS06), ISBN: 1-4244-0196-8, pp. 432-437, London, September 2006.
- [Marrón07a] M. Marrón, J.C. García, M.A. Sotelo, D. Pizarro, I. Bravo. "Extraction of 3D features from complex environments in visual tracking applications", Proceedings of the 2007 IEEE Instrumentation and Measurement Technology Conference (IMTC07), ISBN: 1-4244-1080-0, Warsaw, May 2007.
- [Marrón07b] M. Marrón, M.A. Sotelo, J.C. García, J. Brodfelt. "Comparing improved versions of 'K-Means' and 'Subtractive' clustering in a tracking

- applications”, Proceedings of the Eleventh International Workshop on Computer Aided Systems Theory, Extended Abstracts (EUROCAST07), Lecture Notes in Computer Science, ISBN: 978-84-690-3603-7, Vol. 4739, pp. 717-724, Las Palmas de Gran Canaria, February 2007.
- [Marrón07c] M. Marrón, J.C. García, M.A. Sotelo, F. Huerta, M. Cabello, J. Cerro. “Tracking multiple objects using a Kalman filter and a probabilistic association process”, Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE07), ISBN: 1-4244-0755-9, pp. 2135-2138, Vigo, June 2007.
- [Marrón07d] M. Marrón, J.C. García, M.A. Sotelo, M. Cabello, D. Pizarro, F. Huerta, J. Cerro. “Comparing a Kalman filter and a particle filter in a multiple objects tracking application”, Proceedings of the 2007 IEEE International Symposium on Intelligent Signal Processing (WISP07), ISBN: 1-4244-0829-6. pp. 741-746, Alcalá de Henares, October 2007.
- [Maxwell99] B. Maxwell, L.A. Meeden, N. Addo, L. Brown, P. Dickson, J. Ng, S. Olshfski, E. Silk, J. Wales. “ALFRED: The robot waiter who remembers you”, Proceedings of the AAI Workshop on Robotics, Orlando, July 1999.
- [Maybeck79] P.S. Maybeck. “Stochastic models, estimation and control”. Vol. 1, Academic Press, ISBN: 0-12-480701-01, New York, 1979.
- [Merwe01] R. Van der Merwe, N. de Freitas, A. Doucet, E. Wan. “The unscented particle filter”, Advances in Neural Information Processing Systems, Vol. 13, pp. 584-590, November 2001.
- [MINERVA] MINERVA Carnegie Mellon’s robotic tour-guide robot. <http://www-2.cs.cmu.edu/~minerva/>
- [Moore01] A. Moore. “K-means and hierarchical clustering”, Tutorial 412-268-7599, School of Computer Science, Carnegie Mellon University, November 2001. <http://www.cs.cmu.edu/~awm/tutorials>
- [Ng06] W. Ng, J. Li, S. Godsill. “Online multisensor-multitarget detection and tracking”, Proceedings of the 2006 IEEE Aerospace Conference, ISBN: 0-7803-9545-X, Montana, March 2006.
- [Nicoud96] J.D. Nicoud, Machler P. “Robots for anti-personnel mine search”, Control Engineering Practice, Vol. 4, n° 4, pp. 493-498. April 1996.
- [NURSEBOT] Nursebot Project. <http://www-2.cs.cmu.edu/~nursebot/>
- [Ocaña05] M. Ocaña. “Sistema de localización global WIFI aplicado a la navegación de un robot semi-autónomo”, Tesis Doctoral, Universidad de Alcalá, 2005.

- [Odobez04] J.M. Odobez, D. Gatica-Pérez. “Embedding motion model-based stochastic tracking”, Proceedings of the Seventeenth International Conference on Pattern Recognition (ICPR04), Vol. 2, pp. 815-818, Cambridge, August 2004.
- [Odobez06] J.M. Odobez, D. Gatica-Pérez, S.O. Ba. “Embedding motion in model-based stochastic tracking”, IEEE Transactions on Image Processing, Vol. 15, n° 11, pp. 3515-3531, November 2006.
- [Ogata95] K. Ogata. “Discrete time control systems”, Prentice Hall, ISBN: 0-13-216102-8, New Jersey, 1995.
- [Okuma04] K. Okuma, A. Taleghani, N. De Freitas, J.J. Little, D.G. Lowe. “A boosted particle filter: multitarget detection and tracking”, Proceedings of the Eighth European Conference on Computer Vision (ECCV04), Lecture Notes in Computer Science, ISBN: 3-540-21984-6, Vol. 3021, Part I, pp. 28-39, Prague, May 2004.
- [PATH] PATH Project. <http://www.path.berkeley.edu/>
- [Pelleg00] D. Pelleg, A. Moore. “X-means: Extending k-means with efficient estimation of the number of clusters”, Proceedings of the Seventeenth International Conference on Machine Learning (ICML00), pp. 727-734, ISBN: 1-55860-707-2, San Francisco, 2000.
- [Peñin98] L.F. Peñin, C. Balaguer, J.M. Pastor, F.J. Rodríguez, A. Barrientos, R. Aracil. “Robotized spraying of prefabricated panels”, IEEE Robotics & Automation Magazine, Vol. 5, n° 3, pp. 18-29, September 1998.
- [Pérez02] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. “Color-based probabilistic tracking”, Proceedings of the Seventh European Conference on Computer Vision (ECCV02), Lecture Notes in Computer Science, ISBN: 3-540-43745-2, Vol. 2350, Part I, pp. 661-675, Copenhagen, May 2002.
- [Pérez04] P. Pérez, J. Vermaak, A. Blake. “Data fusion for visual tracking with particles”, Proceedings of the IEEE, Vol. 92, n° 3, pp. 495-513, March 2004.
- [Pineau03] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, S. Thrun. “Towards robotic assistants in nursing homes: Challenges and results”, Robotics and Autonomous Systems, Vol. 42, n° 3, pp. 271–281, March 2003.
- [Pitt99] M.K. Pitt, N. Shephard. “Filtering via simulation: Auxiliary particle filters”, Journal of the American Statistical Association, Vol. 94, n° 446, pp. 590-599, 1999.
- [Pizarro08] D. Pizarro, M. Marrón, D. Peón, M. Mazo, J.C. García, M.A. Sotelo, E. Santiso. “Robot and obstacles localization and tracking with an external

- camera ring”. Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA08), ISBN: 978-1-4244-1647-9, pp. 516-521, Pasadena, May 2008.
- [Pollack02] M. Pollack et al. “Pearl: A mobile robotic assistant for the elderly”, Proceedings of the AAAI Workshop on Automation as Eldercare, Edmonton, August 2002.
- [Prassler01] E. Prassler, J. Scholz, P. Fiorini. “A robotics wheelchair for crowded public environment”, IEEE Robotics & Automation Magazine, Vol. 8, n° 1, pp. 38-45, March 2001.
- [Rasmussen01] C. Rasmussen, G.D. Hager. “Probabilistic data association methods for tracking complex visual objects”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, n°6, pp. 560-576, June 2001.
- [Reid79] D.B. Reid, “An algorithm for tracking multiple targets”, IEEE Transactions on Automatic Control, Vol. 24, n° 6, pp. 843-854, December 1979.
- [Reisman04] P. Reisman, O. Mano, S. Avidan, A. Shashua. “Crowd detection in video sequences”, Proceedings of the IEEE Intelligent Vehicle Symposium (IV04), ISBN: 0-7803-8310-9, pp. 66-71, Parma, June 2004.
- [RHINO] RHINO, Museum Tour-Guide project.
<http://www.informatik.uni-bonn.de/~rhino/tourguide/>
- [Ristic04] B. Ristic, S. Arulampalam, N. Gordon. “Beyond the Kalman filter: Particle filters for tracking applications”, Artech House, ISBN: 978-1-58053-631-8, Boston, 2004.
- [Roecker93] J.A. Roecker, G.L. Phillis. “Suboptimal joint probabilistic data association”, IEEE Transactions on Aerospace and Electronic Systems, Vol. 29, n° 2, pp. 510-517, April 1993.
- [Roy00] N. Roy, G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Margaritis, M. Montemerlo, J. Pineau, J. Schulte, S. Thrun. "Towards personal service robots for the elderly", Proceedings of the Workshop on Interactive Robots and Entertainment (WIRE 2000), Pittsburgh, April 2000.
- [Rui01] Y. Rui, Y. Chen. “Better proposal distributions object tracking using unscented particle filter”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR01), Vol. 2, pp. 786-793, Hawaii, December 2001.
- [Ruspini69] E.H. Ruspini. “A New approach to clustering”, Information Control, Vol. 15, pp. 22-32. 1969.

- [Sarkka07] S. Sarkka, A. Vehtari, J. Lampine. “Rao-Blackwellized particle filter for multiple target tracking”, *Information Fusion Journal*, Vol. 8, n° 1, pp. 2-15, January 2007.
- [Schmitt02] T. Schmitt, M. Beetz, R. Hanek, S. Buck. “Watch their moves applying probabilistic multiple object tracking to autonomous robot soccer”, *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI02)*, ISBN: 0-262-51129-0, pp. 599-604, Edmonton, July 2002.
- [Schraft94] R.D. Schraft. “Mechatronics and robotics for service applications”, *IEEE Robotics & Automation Magazine*, Vol. 1, n° 4, pp. 31-35, 1994.
- [Schulz01] D. Schulz, W. Burgard, D. Fox, A.B. Cremers. “Tracking multiple moving objects with a mobile robot”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR01)*, ISBN: 0-7695-1272-0, Vol. 1, pp. 371-377, Hawaii, December 2001.
- [Schulz03a] D. Schulz, W. Burgard, D. Fox, and AB Cremers. “People tracking with anonymous and ID-sensors using Rao-Blackwellised particle filters”, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI03)*, Acapulco, August 2003.
- [Schulz03b] D. Schulz, W. Burgard, D. Fox, A.B. Cremers. “People tracking with mobile robots using sample-based joint probabilistic data association filters”, *International Journal of Robotics Research*, Vol. 22, n° 2, pp. 99-116, February 2003.
- [Simmons95] R. Simmons, S. Koenig. “Probabilistic navigation in partially observable environments”, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI95)*, pp. 1080-1087, Montreal, August 1995.
- [Smith05] K. Smith, D. Gatica-Pérez, J.M. Odobez. “Using particles to track varying numbers of interacting people”, *Proceedings of the Fourth IEEE Conference on Computer Vision and Pattern Recognition (CVPR05)*, ISBN: 0-7695-2372-2, Vol. 1, pp. 962-969, San Diego, June 2005.
- [Sotelo01] M.A. Sotelo. “Sistema de navegación global aplicado al guiado de un vehículo autónomo terrestre en entornos exteriores parcialmente conocidos”, *Tesis Doctoral*, Universidad de Alcalá, 2001.
- [Spengler03] M. Spengler, B. Schiele. “Multi-Object Tracking Based on a Modular Knowledge Hierarchy”, *Proceedings of the Third International Conference on Computer Vision Systems (ICVS03)*, *Lecture Notes in Computer Science*, Vol. 2626, pp. 376-385, Graz, April 2003.

- [Stentz02] A. Stentz, C. Dima, C. Wellington, H. Herman, D. Stager. "A system for semi-autonomous tractor operations", *Autonomous Robots*, Vol. 13, n° 1, pp. 87-103, July 2002.
- [Sullivan01] J. Sullivan, J. Rittscher. "Guiding Random Particles by Deterministic Search", *Proceedings of the Eighth IEEE International Conference on Computer Vision, (ICCV01)*, Vol. 1, pp. 323-330, Vancouver, July 2001.
- [Tao99] H. Tao, H.S. Sawhney, R. Kumar. "A sampling algorithm for tracking multiple objects", *Proceedings of the International Workshop on Vision Algorithms at (ICCV99)*, *Lecture Notes in Computer Science*, ISBN: 3-540-67973-1, Vol. 1883, pp. 53-68, Corfu, September 1999.
- [Thrun00] S. Thrun. "Probabilistic algorithms in robotics", *Artificial Intelligent Magazine*, Vol. 21, n° 4, pp. 93-109, April 2000.
- [Torma03] P. Torma, C. Szepesvári. "Sequential importance sampling for visual tracking reconsidered", *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, ISBN: 0-9727358-0-1, Key West, January 2003.
- [TOURBOT] [TOURBOT Home Page. http://www.ics.forth.gr/tourbot/](http://www.ics.forth.gr/tourbot/)
- [Trahanias05] P. Trahanias, W. Burgard, A. Argyros, D. Häehnel, H. Baltzakis, P. Pfaff, C. Stachniss. "TOURBOT and WebFAIR: Web-Operated Mobile Robots for Tele-Presence in Populated Exhibitions", *IEEE Robotics & Automation Magazine*, Vol. 12, n°2, pp. 77-89, September 2005.
- [TRILOBITE] [Trilobite 2.0. http://trilobite.electrolux.se/](http://trilobite.electrolux.se/)
- [Tweed02] D. Tweed, A. Calway, "Tracking Many Objects Using Subordinated Condensation", *Proceedings of the British Machine Vision Conference (BMVC02)*, ISBN: 1901725197, pp. 283-292, Cardiff, October 2002.
- [Vermaak03] J. Vermaak, A. Doucet, P. Perez. "Maintaining multimodality through mixture tracking", *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV03)*, ISBN: 0-7695-1950-4, Vol. 2, pp. 1110-1116, Nice, June 2003.
- [Vermaak05a] J. Vermaak, P. Pérez. "Monte-Carlo filtering for multitarget tracking and data association", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 41, n° 1, pp. 309-332, January 2005.
- [Vermaak05b] J. Vermaak, S. Maskell, M. Briers. "A unifying framework for multitarget tracking and existence", *Proceedings of the Eight International Conference on Information Fusion (FUSION05)*, ISBN: 0-7803-9286-8, Vol. 1, Philadelphia, July 2005.

- [Viola01] P. Viola, M. Jones. "Rapid object detection using a boosted cascade of simple features", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR01), ISBN: 0-7695-1272-0, Vol. 1, pp. 511–518, Hawaii, December 2001.
- [Wang96] H.H. Wang, S.M. Rock, M.J. Lee. "OTTER The design and development of an intelligent underwater robot", Autonomous Robots, Vol. 3, n° 2-3, pp. 297-320, June 1996.
- [Weiss97] Y. Weiss. "Belief propagation and revision in networks with loops", Technical Report 1616, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, 1997.
- [Wilson90] R. Wilson, M. Spann. "A new approach to clustering", Pattern Recognition, Vol. 23, n° 12, pp. 1413-1425, 1990.
- [Wright03] R. Wright, S.R. Maskell, M. Briers. "Comparison of Kalman-based methods with particle filters for raid tracking", Proceedings of the Practical Bayesian Statistics Fifth Conference, Milton Keynes, July 2003.
- [Yager94] R.R. Yager, D. Filev. "Generation of fuzzy rules by mountain clustering", Journal of Intelligent and Fuzzy Systems, Vol. 2, n° 3, pp. 209-219, September 1994.
- [Yang03] D.B. Yang, H.H. González-Baños, L.J. Guibas. "Counting People in Crowds with a Real-Time Network of Simple Image Sensors", Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV03), ISBN: 0-7695-1950-4, Vol. 1, pp. 122-129, Nice, October 2003.
- [Zhao04] T. Zhao, R. Nevatia. "Tracking multiple humans in crowded environment", Proceedings of the Third IEEE Conference on Computer Vision and Pattern Recognition (CVPR04), ISBN: 0-7695-2158-4, Vol. 2, pp. 406-413, Washington, June 2004.