

**UNIVERSIDAD DE ALCALÁ**  
**ESCUELA POLITÉCNICA SUPERIOR**  
**DEPARTAMENTO DE ELECTRÓNICA**



**SEGMENTACIÓN Y POSICIONAMIENTO 3D DE ROBOTS  
MÓVILES EN ESPACIOS INTELIGENTES MEDIANTE REDES  
DE CÁMARAS FIJAS**

**TESIS DOCTORAL**

**AUTORA:**

Cristina Losada Gutiérrez

**DIRECTORES:**

Dr. Manuel Mazo Quintas

Dra. Sira Elena Palazuelos Cagigas

2010



# Resumen

La presente tesis doctoral surge con el objetivo de realizar contribuciones para la segmentación, identificación y posicionamiento 3D de múltiples robots móviles. Para ello se utiliza un conjunto de cámaras calibradas y sincronizadas entre sí, que se encuentran ubicadas en posiciones fijas del espacio en que se mueven los robots (espacio inteligente). No se contará con ningún conocimiento a priori de la estructura de los robots móviles ni marcas artificiales a bordo de los mismos.

Tanto para la segmentación de movimiento como para la estimación de la posición 3D de los robots móviles se propone una solución basada en la minimización de una función objetivo, que incorpora información de todas las cámaras disponibles en el espacio inteligente. Esta función objetivo depende de tres grupos de variables: los contornos que definen la segmentación sobre el plano imagen, los parámetros de movimiento 3D (componentes de la velocidad lineal y angular en el sistema de referencia global) y profundidad de cada punto de la escena al plano imagen. Debido a que la función objetivo depende de tres grupos de variables, para su minimización se emplea un algoritmo *greedy*, iterativo, en tres etapas. En cada una de estas etapas dos de los grupos de variables se suponen conocidos, y se resuelve la ecuación para obtener el restante.

De forma previa a la minimización se realiza la inicialización tanto las curvas que definen los contornos de la segmentación como de la profundidad de cada punto perteneciente a los robots. Además se requiere la estimación del número de robots presentes en la escena. Partiendo de que las cámaras se encuentran en posiciones fijas del espacio inteligente, la inicialización de las curvas se lleva a cabo comparando cada imagen de entrada con un modelo de fondo obtenido previamente. Tanto para el modelado de fondo, como para la comparación de las imágenes de entrada con el mismo se emplea el Análisis de Componentes Principales Generalizado (GPCA). Respecto a la

profundidad, se emplea *Visual Hull* 3D (VH3D) para relacionar la información de todas las cámaras disponibles, obteniendo un contorno aproximado de los robots móviles en 3D. Esta reconstrucción de los robots proporciona una buena aproximación de la profundidad inicial de todos los puntos pertenecientes a los robots. Por otro lado, el uso de una versión extendida de la técnica de clasificación *k*-medias permite obtener una estimación del número de robots presentes en la escena.

Tras la segmentación de movimiento y la estimación de la posición 3D de todos los objetos móviles presentes en la escena, se procede a la identificación de los robots móviles. Esta identificación es posible debido a que los robots móviles son agentes controlados por el espacio inteligente, de forma que se cuenta con información acerca de las medidas de los sensores odométricos a bordo de los mismos. Para el seguimiento se propone el uso de un filtro de partículas extendido con proceso de clasificación (XPFCP). La elección de este estimador se debe a que, dado su carácter multimodal, permite el seguimiento de un número variable de elementos (robots móviles) empleando para ello un único estimador, sin necesidad de incrementar el vector de estado. Los resultados obtenidos a la salida del XPFCP son una buena estimación de la posición de los robots móviles en un instante posterior, por lo que esta información se realimenta a la etapa de inicialización de variables, permitiendo reducir el tiempo de procesamiento consumido por la misma.

Las diferentes soluciones propuestas a lo largo de la tesis han sido validadas de forma experimental utilizando para ello diferentes secuencias de imágenes (con presencia de diferentes robots, personas, diversos objetos, cambios de iluminación, etc.) adquiridas en el espacio inteligente del Departamento de Electrónica de la Universidad de Alcalá (ISPACE-UAH).



# Abstract

The main objective of this PhD thesis is to make contributions to motion segmentation and 3D localization of multiple mobile robots in an intelligent space using a multi-camera sensor system. In order to achieve this goal, a set of calibrated and synchronized cameras placed in fixed positions within the environment (intelligent space) are used. Moreover, the proposal in this thesis does not rely on previous knowledge about the robot structure or invasive landmarks on board the robots.

The proposed solution for motion segmentation and 3D localization is based on the minimization of an objective function that combines information from all the cameras. This objective function depends on three groups of variables: the segmentation boundaries, the motion parameters (linear and angular velocities related to the world coordinate system) and the depth of each point in the scene. Since the objective function depends on three groups of variables, its minimization is carried out using a greedy iterative algorithm that consists on three steps that, after initialization of segmentation boundaries and depth, are repeated until convergence. In each of these steps, two of the three groups of variables are fixed, and the equation is solved for the remaining one.

Before the minimization, it is necessary to initialize the curves that define the contours of the segmentation and the depth of each point that belongs to the mobile robots in the images acquired by each camera. Moreover, an estimation of the number of mobile robots included in the scene is required. Since cameras are located in fixed positions within the intelligent space, curve initialization is carried out by comparing each input image to a background model computed previously. Both, the background modelling and the comparison of each input image to the background model are carried out using GPCA. Regarding depth, Visual Hull 3D (VH3D) is used to relate the information from all the available cameras, obtaining a 3D reconstruction of the mobile robots in the world reference system. This 3D reconstruction provides an effective approximation for the initial depth of each point in each camera. Finally, an extended

version of the  $k$ -means algorithm is used to estimate the number of mobile robots in the scene.

After the motion segmentation and 3D position estimation of all mobile objects in the scene, the identification of the robots is obtained. It is possible because the mobile robots are controlled by the intelligent space, and the information of the odometry sensors onboard each robot is available. After that, in order to track the mobile robots, an eXtended Particle Filter with Classification Process (XPFCP) is used. The choice of XPFCP against the basic particle filter (PF) or the extended particle filter (XPF) is due to its multimodal nature. The XPFCP allows tracking a variable number of robots with a single estimator, without increasing the state vector. The results obtained from the XPFCP provide a good estimation for the position of each robot in the next image that is incorporated into the initialization step, reducing the processing time it requires.

In order to validate the solutions proposed in this thesis, several experimental tests have been carried out. In these experimental tests, several image sequences acquired in the intelligent space of the University of Alcalá (ISPACE-UAH) are used. These sequences have different features (different numbers of robots, people walking, objects inserted, illumination changes, etc.) in order to validate the solution under different conditions.

A mi madre, mi hermano y Óscar



# Agradecimientos

La realización de esta tesis ha implicado mucho tiempo y trabajo, así como a numerosas personas. Aunque no es posible nombrarlos a todos aquí deseo, al menos, mencionar a los que han sido más determinantes para el desarrollo de la tesis.

En primer lugar, quiero dar las gracias a mis directores de tesis, Manuel Mazo y Sira Palazuelos por su ayuda, dedicación y paciencia. Sin ellos no hubiera sido posible llegar hasta aquí.

Por otro lado, tengo que agradecer a muchos compañeros su colaboración a diferentes niveles. A Daniel Pizarro y Marta Marrón por sus ideas y consejos, por su ayuda para la implementación de distintos algoritmos. Y por supuesto, a Francisco Redondo, por su ayuda en la programación de los algoritmos, y la realización de pruebas experimentales.

También me gustaría dar las gracias a la gente de la Universidad de Mälardalen en Suecia, especialmente a Mikael Ekström, Nikola Petrovic y Tommy Henriksson.

Aunque no puedo nombrar a cada uno de ellos, no me olvido de mis compañeros del fondo 31 y 22, ellos han hecho que la tesis sea mucho más llevadera.

Entrando en lo personal, quiero dar las gracias a mi madre y mi hermano, por su paciencia y apoyo. También tengo que agradecer enormemente a Óscar por sus ánimos y su ayuda, y por estar ahí siempre que lo he necesitado. Y a mis amigos, que siguen siéndolo a pesar de las ausencias.

Por último, me gustaría hacer extensivo este agradecimiento a todos los compañeros del Departamento de Electrónica de la UAH, alumnos, profesores y PAS por su ayuda a todos los niveles.

A todos ellos, gracias por hacer posible este trabajo.



# Índice de Contenidos

Capítulo 1 Introducción.....	1
1.1. Objetivos generales de la tesis.....	4
1.2. Esquema general del sistema propuesto .....	6
1.3. Estructura de la tesis.....	8
Capítulo 2 Revisión de conocimientos .....	11
2.1. Introducción.....	13
2.2. Segmentación de movimiento .....	19
2.2.1. Técnicas de segmentación de movimiento basadas en diferencia de imágenes.....	20
2.2.2. Técnicas de segmentación de movimiento basadas en flujo óptico ..	22
2.2.3. Otras técnicas de segmentación de movimiento.....	26
2.2.3.1. Basadas en capas ( <i>layers</i> ).....	26
2.2.3.2. Basadas en factorización .....	27
2.3. Posicionamiento de robots móviles a partir de las imágenes adquiridas por cámaras externas.....	28
2.3.1. Posicionamiento utilizando marcas artificiales .....	29
2.3.2. Posicionamiento utilizando marcas naturales.....	33
2.4. Conclusiones.....	35
Capítulo 3 Espacio Inteligente (ISPACE-UAH).....	37
3.1. Estructura del espacio inteligente.....	40
3.1.1. Arquitectura hardware del ISPACE-UAH .....	41
3.1.2. Arquitectura software de ISPACE-UAH.....	42
3.2. Sistemas de referencia de coordenadas en el ISPACE-UAH.....	42
3.2.1. Geometría de formación de la imagen: modelo de cámara pin-hole 44	
3.2.1.1. Parámetros intrínsecos.....	46
3.2.1.2. Parámetros extrínsecos .....	48
Capítulo 4 Segmentación de movimiento y posicionamiento 3D mediante la minimización de una función objetivo .....	51

4.1.	Segmentación de movimiento y posicionamiento 3D a partir de las imágenes adquiridas por una cámara .....	54
4.1.1.	Restricción de brillo 3D para objetos rígidos .....	54
4.1.2.	Función objetivo para una cámara .....	57
4.1.3.	Minimización de la función objetivo .....	59
4.1.3.1.	Etapas del proceso de minimización .....	60
4.1.3.2.	Criterio de convergencia .....	64
4.1.4.	Fusión de los resultados obtenidos para múltiples cámaras .....	66
4.2.	Segmentación de movimiento y posicionamiento 3D mediante la minimización de una función objetivo definida para múltiples cámaras .....	69
4.2.1.	Restricción de brillo 3D para $n_c$ cámaras .....	70
4.2.2.	Función objetivo para $n_c$ cámaras .....	75
4.2.3.	Minimización de la función objetivo .....	77
4.2.3.1.	Etapas del proceso de minimización .....	77
4.2.3.2.	Criterio de convergencia .....	82
4.3.	Conclusiones y aportaciones .....	83
Capítulo 5 Inicialización de las curvas que definen los contornos de los robots móviles .....		85
5.1.	Obtención del modelo de fondo de la escena .....	89
5.1.1.	Elección del número de imágenes y de autovectores utilizados en la obtención del modelo de fondo usando GPCA .....	91
5.1.2.	Comparación entre las técnicas PCA y GPCA para la obtención del modelo de fondo .....	95
5.2.	Selección de píxeles candidatos a pertenecer a robots móviles .....	98
5.2.1.	Cálculo del error de recuperación .....	100
5.2.2.	Umbral del error de recuperación .....	106
5.2.2.1.	Umbral proporcional al valor máximo del error de recuperación .....	107
5.2.2.2.	Umbral adaptativo .....	131
5.3.	Eliminación de sombras .....	143
5.3.1.	Obtención del ángulo de proyección .....	146
5.3.2.	Detección y eliminación de sombras .....	148
5.3.2.1.	Ajuste de los umbrales para la detección y eliminación de sombras .....	150
5.3.3.	Inicialización de las curvas sobre las imágenes invariantes a la iluminación .....	156
5.4.	Conclusiones y aportaciones .....	157
Capítulo 6 Inicialización de la profundidad y estimación del número de robots .....		159
6.1.	Inicialización de la profundidad .....	161
6.1.1.	Visual Hull 3D: definiciones y notación .....	163



6.1.2.	Construcción del Visual Hull .....	164
6.1.3.	Inicialización de la profundidad usando Visual Hull 3D .....	166
6.1.4.	Eliminación de sombras en la reconstrucción 3D .....	167
6.2.	Estimación del número de robots móviles.....	172
6.2.1.	Algoritmo de clustering.....	173
6.2.1.1.	Nomenclatura.....	173
6.2.1.2.	Algoritmo k-medias.....	174
6.2.2.	Estimación del número de robots móviles sobre el plano imagen .	178
6.2.2.1.	Espacio de características .....	179
6.2.2.2.	Medida de la distancia .....	179
6.2.2.3.	Ajuste del valor de la distancia límite .....	181
6.2.3.	Estimación del número de robots móviles en la rejilla de ocupación de <i>Visual Hull 3D</i> .....	186
6.3.	Conclusiones y aportaciones .....	190
Capítulo 7 Función de preservación de bordes y ajuste de los valores de las constantes de ponderación de la función objetivo.....		
7.1.	Elección de la función de preservación de bordes.....	195
7.2.	Valores de las constantes de ponderación $\lambda$ y $\mu$ para la función objetivo definida para una cámara.....	200
7.2.1.	Ajuste de la constante $\mu$ .....	200
7.2.1.1.	Efecto del valor de la constante $\mu$ sobre las curvas que definen los contornos de las regiones en que se segmenta la imagen .....	201
7.2.1.2.	Efecto del valor de la constante $\mu$ sobre la profundidad. .	201
7.2.1.3.	Efecto del valor de la constante $\mu$ sobre las componentes de la velocidad 3D en el sistema de referencia de la cámara	203
7.2.2.	Ajuste de la constante $\lambda$ .....	207
7.2.2.1.	Efecto del valor de la constante $\lambda$ sobre la profundidad..	207
7.2.2.2.	Efecto del valor de la constante $\lambda$ sobre las curvas que definen las regiones en las que se segmenta la imagen ...	209
7.2.2.3.	Efecto del valor de la constante $\lambda$ sobre la velocidad 3D	210
7.2.3.	Validación de los valores de $\lambda$ y $\mu$ elegidos.....	212
7.3.	Valores de las constantes $\lambda$ y $\mu$ para la función objetivo definida para múltiples cámaras .....	214
7.3.1.	Ajuste de la constante $\mu$ .....	214
7.3.1.1.	Efecto del valor de la constante $\mu$ sobre las curvas que definen las regiones en que se segmenta la imagen.....	216
7.3.1.2.	Efecto del valor de la constante $\mu$ sobre la profundidad .	217
7.3.1.3.	Efecto del valor de la constante $\mu$ sobre la velocidad 3D y el número de iteraciones hasta la convergencia .....	218
7.3.2.	Ajuste de la constante $\lambda$ .....	220

7.3.2.1.	Efecto del valor de la constante $\lambda$ sobre la profundidad..	221
7.3.2.2.	Efecto del valor de la constante $\lambda$ sobre las curvas que definen las regiones en que se segmenta la imagen.....	222
7.3.2.3.	Efecto del valor de la constante $\lambda$ sobre la velocidad 3D y el número de iteraciones hasta la convergencia .....	224
7.3.3.	Validación de los valores de $\lambda$ y $\mu$ elegidos .....	225
7.4.	Conclusiones y aportaciones .....	227
Capítulo 8	Identificación y seguimiento de múltiples robots móviles .....	229
8.1.	Identificación de los robots móviles.....	232
8.2.	Filtro de partículas extendido con proceso de clasificación (XPFCP) .....	233
8.2.1.	Filtro de partículas extendido (XPF) .....	234
8.2.2.	Clasificador.....	237
8.3.	XPFCP para el seguimiento de múltiples robots móviles .....	238
8.3.1.	Modelo del sistema.....	239
8.3.2.	Seguimiento mediante XPFCP .....	240
8.3.2.1.	Implementación del XPFCP .....	241
8.3.2.2.	Clasificador.....	242
8.3.3.	Realimentación de la salida del XPFCP al módulo de inicialización de variables.....	243
8.4.	Conclusiones y aportaciones .....	246
Capítulo 9	Resultados y validación de la solución propuesta .....	249
9.1.	Descripción de las secuencias de imágenes utilizadas en la evaluación ....	251
9.2.	Inicialización de variables .....	255
9.2.1.	Inicialización de las curvas.....	256
9.2.2.	Inicialización de la profundidad .....	261
9.3.	Segmentación de movimiento y estimación de la posición 3D de los robots móviles.....	264
9.4.	Seguimiento de múltiples robots móviles.....	270
9.4.1.	Efecto del número de partículas .....	272
9.5.	Validación del sistema completo.....	275
9.6.	Conclusiones.....	286
Capítulo 10	Resumen, conclusiones y aportaciones.....	287
10.1.	Resumen .....	289
10.1.1.	Inicialización de variables .....	290
10.1.2.	Segmentación de movimiento y posicionamiento 3D.....	293
10.1.3.	Identificación y seguimiento de los robots móviles .....	294
10.2.	Conclusiones.....	294

10.3. Aportaciones más relevantes .....	298
10.4. Líneas de trabajo futuro .....	300
10.5. Publicaciones derivadas de la tesis .....	301
Referencias .....	303



# Índice de Figuras

Figura 1.1. Estructura general del Espacio Inteligente en el que se representan los diferentes elementos del mismo.....	5
Figura 1.2. Esquema general de la solución propuesta en esta tesis. Los bloques representados en color azul representan las diferentes tareas desarrolladas para la fase de segmentación y obtención del posicionamiento. ....	7
Figura 2.1. Imagen perteneciente a un video en que se demuestran las capacidades de la “Smart Room” desarrollada por el grupo VisMod del MIT para la interacción entre una persona y una mascota virtual. (Imagen extraída de la página web del proyecto: <a href="http://vismod.media.mit.edu/vismod/demos/smartroom/">http://vismod.media.mit.edu/vismod/demos/smartroom/</a> ).....	14
Figura 2.2. Esquema general del proyecto “Oxygen” desarrollado en el laboratorio de Ciencias de la Computación e Inteligencia Artificial del MIT.....	15
Figura 2.3. Imágenes de la “Smart meeting room” desarrollada por el grupo SAIL de la USC. (a) Imagen general de (b) Detalle del array de micrófonos y la cámara omnidireccional.....	15
Figura 2.4. Esquema de la sala de reuniones implementada en el instituto IDIAP. Imagen extraída de [Hung et al. 2008]. ....	16
Figura 2.5. Esquema de la red de sensores propuesta por Hoover y Olsen [Hoover & Olsen 1999a], y mapa de ocupación sobre el plano del suelo. ....	17
Figura 2.6. Topología de la arquitectura formada por LAPUs y RCUs propuesta en el sistema MEPHISTO.....	17
Figura 2.7. (a) Imágenes del robot en movimiento (b) Resultado de la umbralización de la diferencia del par de imágenes, calculada píxel a píxel (c) Resultado de la umbralización de la diferencia entre cada imagen de entrada y una imagen de referencia en la que únicamente aparece el fondo de la escena.....	21
Figura 2.8. El flujo óptico no siempre se corresponde con el campo de movimiento. A la izquierda se muestra una esfera sin textura, bajo iluminación constante que rota sobre sí misma, sin que exista movimiento en la imagen. A la derecha se muestra una esfera fija, sometida a cambios en la iluminación, que provocan que exista un campo de movimiento (flujo óptico) en el plano imagen. ....	22
Figura 2.9. Representación gráfica del problema de apertura en el cálculo del flujo óptico. ....	24
Figura 2.10. Ejemplo de las marcas artificiales utilizadas para la navegación de robots móviles (a) Marca formada por rectángulos simétricos y dos números de 7 segmentos [Yang & Xinhe 2006] (b) marca triangular empleada el trabajo de [Sooyong & Jae-Bok 2007] (c) Código MR ( <i>Mobile Robot</i> ) definido en [Feng et al 2009]. ....	31
Figura 2.11. Marca artificial empleada en [Zhang et al. 2004] para la localización de un robot bajo el agua .....	31

Figura 2.12. Marcas artificiales empleadas sobre un robot móvil para su localización e identificación (a) marca utilizada en [Fan et al. 2008]. (b) marca utilizada en [Liang et al. 2008].	32
Figura 2.13. Array de diodos de infrarrojo empleados para el posicionamiento y guiado de un robot móvil en [Fernandez et al. 2007].	32
Figura 2.14. Esquema de funcionamiento del sistema de guiado de un dirigible mediante marcas artificiales descrito en [Coelho & Campos 1999].	33
Figura 3.1. Espacio inteligente del Departamento de Electrónica de la UAH (ISPACE-UAH).	39
Figura 3.2. Estructura general del ISPACE-UAH considerado en esta tesis en la que se muestran los diferentes elementos existentes en el mismo.	40
Figura 3.3. Diagrama general de la arquitectura hardware y software del ISPACE-UAH.	42
Figura 3.4. Sistemas de referencia en el ISPACE-UAH. Los ejes de coordenadas del sistema de referencia del mundo ( $\Gamma_w$ ) se representan en color rojo, mientras que los ejes del sistema de referencia de cada cámara ( $\Gamma_{ci}$ , $i = 1, 2, \dots, n_c$ ) se representan en color azul, y los ejes del sistema de referencia del robot en color verde.	43
Figura 3.5. Sistema de referencia 3D de la cámara $i$ ( $\Gamma_{ci}$ , $i=1, 2, \dots, n_c$ ) en la que el origen de coordenadas se ubica en el centro óptico y el eje $Z_{ci}$ coincide con el eje óptico de la cámara.	44
Figura 3.6. (a) Modelo de formación de la imagen en una cámara <i>pin-hole</i> representada como una caja cerrada con un agujero en uno de sus lados. (b) Proyección equivalente: la proyección invertida de un objeto en el plano imagen a una distancia $f$ del centro óptico de la cámara es equivalente a la proyección de dicho objeto en un plano situado a una distancia $-f$ del centro de la cámara.	45
Figura 3.7. Modelo geométrico de cámara pin-hole. En color azul se representan los ejes del sistema de referencia de la cámara ( $\Gamma_c$ ) mientras que en color negro se muestran los ejes de coordenadas sobre el plano imagen ( $x, y$ ) y en línea roja discontinua la línea de proyección del punto 3D $\mathbf{P}_c(X_c, Y_c, Z_c)$ sobre el plano imagen en el píxel $\mathbf{p}(x, y)$ .	45
Figura 3.8. Relación entre sistemas de referencia. En color negro se representan los ejes del sistema de referencia global (del mundo) $\Gamma_w$ mientras que en azul se muestran los ejes del sistema de referencia de una cámara ( $\Gamma_c$ ).	48
Figura 4.1. Diagrama de bloques general del proceso de minimización de la función objetivo definida en la ecuación (4.14).	60
Figura 4.2. Diagrama de bloques general del proceso de segmentación de movimiento y obtención de la posición 3D de robots móviles a partir de las imágenes captadas por un conjunto de cámaras ubicadas en posiciones fijas en el espacio.	67
Figura 4.3. Diagrama de bloques general del algoritmo de segmentación de movimiento rígido y obtención de la posición 3D a partir de las imágenes captadas por $n_c$ cámaras ubicadas en posiciones fijas del espacio. Tanto la segmentación de movimiento como la obtención de la posición 3D se realizan de forma conjunta para las $n_c$ imágenes, mientras que el proceso de inicialización de las curvas se ejecuta por cada una de las cámaras disponibles.	70
Figura 5.1. Diagrama de bloques general en el que se presentan las diferentes etapas necesarias para la inicialización de las curvas que definen los contornos de los robots móviles presentes en la escena.	88
Figura 5.2. Obtención del modelo de fondo de la escena usando GPCA a partir de un conjunto de imágenes de fondo en las que no aparecen robots móviles. Esta etapa sólo se ejecuta una vez y puede realizarse <i>off-line</i> .	90
Figura 5.3. Error de recuperación de las imágenes de fondo en función del número de imágenes utilizadas para el cálculo del modelo de fondo y utilizando diferente número de autovectores.	93

Figura 5.4. Tiempo consumido en la obtención del modelo de fondo en función del número de imágenes utilizadas para el cálculo, para 1, 2 y 5 autovectores. ....	93
Figura 5.5. (a) y (b) Imágenes del fondo de la escena. (c) y (d) Resultado de la proyección y posterior recuperación de las imágenes (a) y (b) mediante las matrices de transformación GPCA obtenidas a partir de 10 imágenes del fondo con 1 único autovector. (e) y (f) Error de recuperación obtenido como la diferencia entre la imagen original y la recuperada elevada al cuadrado. ....	94
Figura 5.6. Tiempo de cómputo para el cálculo de las matrices de transformación PCA y GPCA (a) en función del número de autovectores y (b) en función del número de imágenes utilizadas. ....	96
Figura 5.7. Error de recuperación de PCA y GPCA (a) En función del número de autovectores, para 10 imágenes de fondo. (b) En función del número de imágenes con 1 autovector. ....	97
Figura 5.8. (a) Imagen original en la que aparece un robot móvil (b) Imagen obtenida tras proyectar la imagen de entrada al espacio transformado (ecuación (5.7)) de dimensión $1 \times 1$ y posteriormente recuperarla en el espacio original (ecuación (5.8)). ....	99
Figura 5.9. Diagrama de bloques para la selección de los píxeles candidatos a pertenecer a robots móviles mediante la umbralización del error de recuperación calculado utilizando la técnica GPCA. ....	100
Figura 5.10. Cálculo del error de recuperación utilizando ventanas. En el caso de esta figura la ventana se ha considerado de $3 \times 3$ píxeles. ....	101
Figura 5.11. Imágenes pertenecientes a las secuencias con las que se han realizado las diferentes pruebas que permiten la evaluación del efecto del tamaño de la ventana utilizado en el cálculo del error de recuperación. ....	102
Figura 5.12. Error de recuperación en función del tamaño de la ventana utilizada para la imagen de la Figura 5.11.a. (a) Sin utilizar ventanas (ecuación (5.9)). (b) Con una ventana de $3 \times 3$ píxeles. (c) Con una ventana de $7 \times 7$ píxeles. (d) Con una ventana de $13 \times 13$ píxeles. ....	103
Figura 5.13. Error de recuperación en función del tamaño de la ventana utilizada, para la imagen de la Figura 5.11.b. (a) Sin utilizar ventanas (ecuación (5.9)). (b) Con una ventana de $3 \times 3$ píxeles. (c) Con una ventana de $7 \times 7$ píxeles. (d) Con utilizar una ventana de $13 \times 13$ píxeles. ....	104
Figura 5.14. Valor del error de recuperación en los píxeles de la columna 150 de la imagen mostrada en la Figura 5.11.a, para diferentes valores del tamaño de ventana. ....	104
Figura 5.15. Valor del error de recuperación en los píxeles de la columna 350 de la imagen mostrada en la Figura 5.11 (b) para diferentes tamaños de ventana. ....	105
Figura 5.16. Tiempo de cómputo consumido en el cálculo del error de recuperación (en segundos) en función del tamaño de la ventana utilizada. En azul se muestra el tiempo consumido por 50 imágenes de la secuencia 1 (Figura 5.11.a). Con línea discontinua roja se presenta el tiempo de cómputo de las imágenes de la secuencia 2 (Figura 5.11.b). ....	105
Figura 5.17. Curvas iniciales para diferentes valores del umbral sobre el error de recuperación (a) Curva obtenida con un umbral demasiado alto. (b) Curva obtenida con un umbral demasiado bajo. (c) Curva obtenida con el umbral adecuado. ....	107
Figura 5.18. Diagrama de bloques general que refleja el proceso de obtención del valor inicial del umbral en función del valor máximo del error de recuperación obtenido para las imágenes implicadas en el cálculo del modelo de fondo. ....	108
Figura 5.19. (a) y (b) Imágenes de entrada. (c) Valor del error de recuperación en los píxeles de la columna $j=150$ marcada en negro en (a) y umbral proporcional (5.13). (d) Valor del error de recuperación en los píxeles de la columna $j=350$ marcada en negro en (b) y umbral proporcional (5.13). (e) y (f) Curvas iniciales obtenidas tras aplicar el umbral proporcional definido en (5.13). ....	109

Figura 5.20. (a) y (b) Imágenes de entrada. (c) Valor del error de recuperación en los píxeles de la columna $j=150$ marcada en negro en (a). (d) Valor del error de recuperación en los píxeles de la columna $j=350$ marcada en negro en (b).....	110
Figura 5.21. (a) Imagen de prueba de la secuencia 1. (b) Resultado obtenido tras umbralizar el error de recuperación de (a) con el valor del umbral óptimo. (c) Imagen de prueba de la secuencia 2. (d) Resultado obtenido tras umbralizar (c) con el valor del umbral óptimo. ....	112
Figura 5.22. (a) y (c) Valor del error de recuperación para los píxeles de una columna de las imágenes mostradas en la Figura 5.20 (a) y (b) y umbrales para los diferentes valores de la constante $K$ considerados. (c) y (d) Detalle de la imagen y las curvas iniciales obtenidas para los diferentes valores de la constante $K$ .....	113
Figura 5.23. Curvas iniciales obtenidas para diferentes imágenes al aplicar sobre el error de recuperación un umbral proporcional (con $K=10$ ) al valor máximo del error de recuperación de las imágenes de fondo (5.14).....	115
Figura 5.24. Valor del umbral proporcional, con $K=10$ , en función del número de imágenes para 1, 2 y 5 autovectores. (a) Imágenes de la secuencia 1 (Figura 5.20 (a)). (b) Imágenes de la secuencia 2 (Figura 5.20 (b)).....	118
Figura 5.25. Valor del umbral proporcional (5.14) con $K=10$ en función del número de imágenes y autovectores considerados en la obtención del modelo de fondo de la escena usando GPCA (a) para las imágenes de la secuencia 1, (b) para las imágenes de la secuencia 2. ....	118
Figura 5.26. Tiempo de cómputo consumido en el cálculo y umbralización del error de recuperación para la obtención de las curvas iniciales en función del número de imágenes y de autovectores considerados en el modelado del fondo de la escena. (a) Secuencia de imágenes 1 (Figura 5.20 (a)). (b) Secuencia de imágenes 2 (Figura 5.20 (b)).....	119
Figura 5.27. Valor del umbral en función del tamaño de la ventana utilizada para el cálculo del error de recuperación.....	120
Figura 5.28. Valor máximo del error de recuperación de las imágenes de fondo en cada píxel para diferentes valores del tamaño de la ventana elegida (a) $q = 1$ píxeles (b) $q = 7$ píxeles.....	121
Figura 5.29. Valor máximo del error de recuperación de las imágenes de fondo en función del tamaño de la ventana para una columna de la imagen.....	121
Figura 5.30. Curvas iniciales obtenidas tras aplicar sobre el error de recuperación el umbral proporcional a $K$ con $K=10$ calculado para un tamaño de ventana de $q = 1, 3$ y $11$ píxeles.....	122
Figura 5.31. Número de píxeles en los que el error de recuperación supera el umbral (Ecuación (5.14)) en función del valor de la constante $K$ y el tamaño de ventana utilizado en el cálculo del error. (a) Imagen de la secuencia 1. (b) Imagen de la secuencia 2.....	123
Figura 5.32. Valores válidos de la constante $K$ en el cálculo del umbral sobre el error de recuperación en función del tamaño de la ventana utilizada para las imágenes de entrada mostradas en la Figura 5.19(a) y (b). ....	123
Figura 5.33. Valor inicial del umbral en función del tamaño de la ventana utilizada para el cálculo del modelo de fondo actualizando el valor de la constante $K$ en función del tamaño de la ventana. .	124
Figura 5.34. Porcentaje de píxeles segmentados de forma correcta en función del tamaño de la ventana utilizada en el cálculo del error de recuperación tras aplicar un umbral definido según la ecuación (5.14) con $K$ constante (en línea azul continua) y con $K$ definida en función del tamaño de ventana según las ecuaciones (5.15) y (5.16), respectivamente. (a) Imagen de la secuencia 1 (Figura 5.19 (a)). (b) Imagen de la secuencia 2(Figura 5.19 (b)).....	125
Figura 5.35. Tiempo de cómputo medio consumido en la obtención de la curva inicial en 50 imágenes en función del tamaño de la ventana utilizada en el cálculo del error de recuperación. ....	126



Figura 5.36. Intensidad media de las $N_i$ imágenes utilizadas en el cálculo del modelo de fondo (en línea roja en la gráfica superior). Intensidad media y número de puntos contenidos por la curva inicial, aplicando el valor inicial del umbral proporcional (5.14) con $K=10$ sobre el error de recuperación de 700 imágenes pertenecientes a la secuencia de la imagen mostrada en la Figura 5.11(a). .....	127
Figura 5.37. Intensidad media de las $N_i$ imágenes utilizadas para el cálculo del modelo de fondo (en línea roja discontinua en la gráfica superior). Intensidad media y número de puntos contenidos por la curva inicial, aplicando el valor inicial del umbral proporcional (5.14) con $K=10$ sobre el error de recuperación de 700 imágenes pertenecientes a la secuencia de la imagen mostrada en la Figura 5.11(b). .....	127
Figura 5.38. Número de puntos contenidos en la curva inicial en función de la intensidad media de la imagen de entrada. Estas gráficas se han obtenido representando para cada imagen el punto correspondiente a su intensidad media, y al número de píxeles en los que el error de recuperación supera el umbral fijado definido en la ecuación (5.14). En (a) se muestran los resultados para 700 imágenes de la secuencia 1 (Figura 5.11 (a)) mientras que en (b) se han representado los resultados para 700 imágenes de la secuencia 2 (Figura 5.11 (b)) .....	128
Figura 5.39. Ejemplo de las imágenes utilizadas para evaluar el efecto de los cambios de iluminación: (a) y (d) imágenes originales. (b) y (e) imágenes obtenidas tras restarle un valor de 40 a la intensidad de todos los píxeles de las imágenes originales. (c) y (f) imágenes obtenidas tras sumar un valor de 40 a la intensidad de todos los píxeles de las imágenes originales. ....	129
Figura 5.40. Efecto de la variación de la intensidad de las imágenes sobre las curvas iniciales. (a) Número de píxeles que contiene la curva inicial en función del valor que se suma a la intensidad en cada píxel de la imagen de entrada para un umbral proporcional con $K = 10$ . (b) Curva inicial obtenida tras restar un valor igual a 70 a la intensidad de los píxeles de la imagen de entrada. (c) Curva inicial obtenida para la imagen de entrada original. (d) Curva inicial obtenida tras sumar un valor de 70 a la intensidad de los píxeles de la imagen de entrada. ....	130
Figura 5.41. Efecto de la constante $\alpha$ sobre las curvas iniciales: número de píxeles contenidos en la curva inicial en función del valor sumado a la intensidad de todos los píxeles de la imagen de entrada para diferentes valores de la constante $\alpha$ con $\beta = 0.75$ . ....	133
Figura 5.42. Efecto de la constante $\beta$ sobre las curvas iniciales: número de píxeles contenidos en la curva inicial en función del valor sumado a la intensidad de todos los píxeles de la imagen de entrada para diferentes valores de la constante $\beta$ con $\alpha = 2.5$ . ....	133
Figura 5.43. (a) Rango de valores en los que es posible incrementar o decrementar la intensidad en cada píxel de la imagen de entrada obteniendo las curvas iniciales de forma correcta en función de los valores de las constantes $\alpha$ y $\beta$ . (b) Detalle del rango de valores para $\alpha$ entre 1 y 11 y $\beta$ entre 1 y 1.8. ....	134
Figura 5.44. Detalle de las curvas iniciales obtenidas tras aplicar sobre el error de recuperación el umbral dinámico obtenido para diferentes valores de las constantes $\alpha$ y $\beta$ . ....	135
Figura 5.45. Curvas iniciales obtenidas aplicando el umbral proporcional (5.14) con $K=10$ y el umbral adaptativo (5.18) al error de recuperación obtenido en imágenes con diferentes valores de intensidad media para una variación homogénea de la intensidad. En rojo se muestra el resultado obtenido al utilizar un umbral fijo (ecuación (5.14)) mientras que en azul se muestran las curvas iniciales obtenidas con el umbral dinámico (ecuación (5.18)). ....	137
Figura 5.46. Curvas iniciales obtenidas aplicando el umbral proporcional (5.14) con $K=10$ (en rojo) y el umbral adaptativo (5.18) (en azul) al error de recuperación obtenido en imágenes a las que se le ha modificado la intensidad de forma no homogénea. ....	138
Figura 5.47. Curvas iniciales obtenidas (a) utilizando una circunferencia según se propone en [Sekkati & Mitiche 2006b], y (b) utilizando la técnica propuesta en esta tesis basada en la umbralización del error de recuperación calculado usando la técnica GPCA. ....	140

Figura 5.48. Imágenes de prueba pertenecientes a las secuencias 3 y 4. (a) Imagen 250 de la secuencia 3. (b) Imagen 300 de la secuencia 3. (c) Imagen 350 de la secuencia 3. (d) Imagen 250 de la secuencia 4. (e) Imagen 300 de la secuencia 4. (f) Imagen 350 de la secuencia 4. ....	141
Figura 5.49. Curvas iniciales obtenidas para las imágenes de prueba de las secuencias 3 y 4 mostradas en la Figura 5.48. (a) Imagen 250 de la secuencia 3. (b) Imagen 300 de la secuencia 3. (c) Imagen 350 de la secuencia 3. (d) Imagen 250 de la secuencia 4. (e) Imagen 300 de la secuencia 4. (f) Imagen 350 de la secuencia 4. ....	141
Figura 5.50. Porcentaje de píxeles segmentados de forma correcta en función del valor sumado a la intensidad de cada píxel de la imagen de entrada. Resultados obtenidos para las imágenes de la secuencia 3 (en rojo en la figura) y la secuencia 4 (en negro en la figura). ....	143
Figura 5.51. Espacio cromático logarítmico 2D. En azul se muestra el plano en el que se define el espacio de color logarítmico 2D $\chi = (\chi_1, \chi_2)$ definido según la ecuación (5.26) a partir del espacio logarítmico de color $\mathbf{p} = (\rho_R, \rho_G, \rho_B)^T$ . ....	145
Figura 5.52. Representación de la obtención del ángulo de proyección mediante la minimización de la entropía del histograma de la imagen proyectada. (a) Ejemplo en el que la proyección se ha realizado utilizando el ángulo correcto. (b) Histograma de la proyección correcta. (c) Ejemplo en el que se ha realizado la proyección utilizando un ángulo incorrecto. (d) Histograma de la proyección incorrecta. ....	146
Figura 5.53. Valor de la entropía del histograma de la imagen proyectada (ecuación (5.27)) en función del ángulo utilizado para la proyección. El ángulo de proyección elegido es aquel para el que la entropía es mínima. ....	147
Figura 5.54. (a) Imagen de entrada en la que se ha representado la región de interés (ROI) mediante un rectángulo en rojo. (b) Detalle de la región de interés en la imagen (ROI). (c) Imagen invariante a la iluminación obtenida a partir de la ROI de la imagen original. ....	148
Figura 5.55. Eliminación de sombras. Curvas de contornos iniciales de robots móviles antes (en rojo) y después (en negro) del proceso de eliminación de sombras. (a) Curvas iniciales obtenidas en el caso de aplicar un umbral $U_{inv}$ adecuado. (b) Curvas iniciales obtenidas en el caso de aplicar un umbral $U_{inv}$ demasiado pequeño. (c) Curvas iniciales obtenidas en el caso de aplicar un umbral $U_{inv}$ demasiado grande. ....	150
Figura 5.56. Imágenes de prueba pertenecientes a la secuencia 1. Se muestran las imágenes que ocupan las posiciones 200, 300 y 400 de esta secuencia. ....	151
Figura 5.57. Imágenes de prueba pertenecientes a la secuencia 2. Se muestran las imágenes que ocupan las posiciones 200, 300 y 400 de esta secuencia. ....	151
Figura 5.58. Valor medio del porcentaje de píxeles segmentados de forma correcta (a) para el conjunto de imágenes de la secuencia 1 (Figura 5.56) y (b) para el conjunto de imágenes de la secuencia 2 (Figura 5.57). ....	152
Figura 5.59. Detalle de las curvas iniciales antes (en rojo) y después (en azul) del proceso de eliminación de sombras propuesto para diferentes imágenes de las secuencias de prueba 1, 2 y 3. ....	154
Figura 5.60. (a) Bordes detectados al umbralizar la norma del gradiente de la imagen original. (b) Bordes detectados al umbralizar la norma del gradiente de la imagen invariante. (c) Resultado de la umbralización utilizando la función definida en la ecuación (5.30). ....	155
Figura 5.61. Diagrama de bloques del proceso de eliminación de los píxeles que pertenecen a sombras en las curvas iniciales. La eliminación de sombras se realiza en un espacio de color invariante a la iluminación. ....	155
Figura 5.62. Propuesta de inicialización de curvas de contornos de robots, utilizando imágenes invariantes a la iluminación. ....	156

Figura 5.63. Ejemplo de imágenes invariantes a la iluminación. Todas las imágenes pertenecen a la secuencia 1 (Figura 5.19 (a)). (a) Imagen de fondo. (b) Imagen de entrada invariante a la iluminación. (c) Imagen de entrada en escala de grises. ....	157
Figura 6.1. Visión general del proceso de inicialización, utilizando Visual Hull 3D de la profundidad a partir de las curvas iniciales que definen los contornos de los posibles robots móviles. ....	162
Figura 6.2. Ejemplo de construcción del <i>Visual Hull</i> a partir de siluetas en 2D. (a) mediante la intersección directa de los conos visuales (el resultado de la intersección se muestra en color verde alrededor del objeto <i>O</i> ). (b) Mediante aproximación basada en vóxeles (la región mostrada en amarillo representa el <i>Visual Hull</i> aproximado, mientras que la línea verde se corresponde con la intersección real de los conos visuales). ....	165
Figura 6.3. Efecto de mejora de las curvas iniciales conseguida tras la inicialización de la profundidad mediante <i>Visual Hull 3D</i> . (a) Rejilla de ocupación obtenida a partir de las imágenes captadas por tres de las cámaras disponibles en el ISPACE-UAH. (b) Curva inicial obtenida para la imagen captada por la cámara 3. (c) Curva inicial mejorada tras la inicialización de la profundidad mediante <i>Visual Hull 3D</i> . Las curvas que aparecían fuera del robot han sido eliminadas. ....	167
Figura 6.4. Ejemplo de imagen perteneciente a la secuencia de test vista por las cuatro cámaras disponibles en el ISPACE-UAH. (a) Imagen de la cámara 1. (b) Imagen de la cámara 2. (c) Imagen de la cámara 3. (d) Imagen de la cámara 4. ....	168
Figura 6.5. Detalle de las curvas de contorno de un robot obtenidas para una imagen de la secuencia de test (Figura 6.4) tras la eliminación de las zonas de sombra. (a) Utilizando <i>Visual Hull 3D</i> y eliminando los puntos con $Z_w=0$ en $\Gamma_w$ . (b) Tras la proyección de la imagen a un espacio invariante a la iluminación. ....	169
Figura 6.6. Detalle de las curvas obtenidas para diferentes imágenes pertenecientes a la secuencia de test (Figura 6.4). En color rojo se representa la curva inicial obtenida mediante GPCA. La curva verde es la obtenida tras la proyección de la imagen de entrada a un espacio invariante a la iluminación, mientras que la azul es el resultado de eliminar las muestras en las que $Z_w=0$ en $\Gamma_w$ . ....	170
Figura 6.7. Comparación del tiempo de cómputo consumido en la eliminación de los puntos de sombra presentes en las curvas iniciales mediante la proyección a un espacio invariante a la iluminación, y la eliminación del plano $Z_w=0$ en $\Gamma_w$ para un conjunto de imágenes pertenecientes a la secuencia de test (Figura 6.4). ....	171
Figura 6.8. Diagrama de bloques general del algoritmo k-medias básico. ....	175
Figura 6.9. Diagrama de bloques general del algoritmo k-medias extendido para permitir la creación de clases nuevas. Los bloques añadidos al algoritmo k-medias básico aparecen en color amarillo. ....	176
Figura 6.10. Resultados de la ejecución del algoritmo <i>k</i> -medias sobre un conjunto de datos <b>Y</b> en un espacio 2D, utilizando como características las coordenadas <i>x,y</i> de los puntos segmentados y la distancia euclídea como criterio de asignación de elementos a clases. (a) Resultado obtenido aplicando el algoritmo <i>k</i> -medias básico con $k=2$ . (b) Resultado obtenido aplicando el algoritmo <i>k</i> -medias básico con $k=3$ . (c) Resultado obtenido aplicando el algoritmo <i>k</i> -medias extendido. Las circunferencias representan el valor de <i>dMax</i> en el espacio de clasificación 2D. ....	177
Figura 6.11. Resultados de la ejecución del <i>k</i> -medias extendido sobre un conjunto de medidas en un espacio bidimensional para diferentes valores del parámetro <i>dMax</i> . (a) $dMax = dMax_0$ las tres clases se detectan correctamente. (b) $dMax = \frac{dMax_0}{1.5}$ una de las clases aparece duplicada. (c) $dMax = 1.5 \cdot dMax_0$ dos de las clases aparecen unidas. ....	178
Figura 6.12. Imágenes pertenecientes a la secuencia 4 (Figura 5.48d) sobre las que se muestran los puntos candidatos a pertenecer a robots móviles, obtenidos tras la inicialización de las curvas mediante GPCA y eliminación de sombras. ....	179

Figura 6.13. Resultados obtenidos tras aplicar $k$ -medias sobre los datos de entrada mostrados en la Figura 6.12 utilizando como medida la distancia euclídea, con un valor de distancia límite $dMax = 150$ .	180
Figura 6.14. Resultado de la proyección sobre el eje $x$ de las medidas mostradas en las imágenes de la Figura 6.12(a) y (b).	181
Figura 6.15. Resultado del clustering tras aplicar $k$ -medias utilizando como medida de distancia la ecuación (6.5) (imágenes mostradas a la izquierda) y la ecuación (6.4) (imágenes mostradas a la derecha) para (a) Imagen 1 de la secuencia.	184
Figura 6.16. Imágenes pertenecientes a la secuencia de prueba.	185
Figura 6.17. Resultado del clustering de las imágenes mostradas en la Figura 6.16.	185
Figura 6.18. Ejemplo de clustering para la detección del número de objetos móviles. (a) Imagen de la escena real captada por una cámara. (b) Conjunto de medidas 3D extraídas de la imagen (a). (c) Proyección sobre el espacio cartesiano 2D definido por el plano XY.	187
Figura 6.19. Resultado del <i>clustering</i> . Las imágenes mostradas en la parte izquierda muestran el resultado de aplicar el algoritmo $k$ -medias sobre el conjunto de muestras proyectadas sobre el plano XY en $\Gamma_w$ , mientras que las imágenes de la columna derecha representan las muestras en 3D en $\Gamma_w$ .	189
Figura 7.1. Imágenes pertenecientes a la secuencia 1, utilizada para el ajuste de las constantes $\lambda$ y $\mu$ y para la elección de la función de preservación de bordes.	194
Figura 7.2. Imágenes pertenecientes a la secuencias 2 y 3, utilizadas para validar los valores elegidos para las constantes de ponderación $\lambda$ y $\mu$ .	194
Figura 7.3. (a) Representación gráfica de las funciones de preservación de bordes presentadas en la Tabla 7.1 para valores de $\ \nabla Z\ $ comprendidos entre 0 y 2. (b) Representación gráfica de las derivadas de las funciones de preservación de bordes divididas entre $\ \nabla Z\ $ .	196
Figura 7.4. Tiempo de cómputo consumido por la minimización de la función objetivo calculado para 100 imágenes, para las diferentes funciones de preservación de bordes consideradas.	199
Figura 7.5. (a) Número de píxeles contenidos en la curva que define el contorno del robot móvil en función del valor de la constante $\mu$ para 10 imágenes de prueba de la secuencia a la que pertenecen las imágenes de la Figura 7.1. (b) Detalle de las curvas obtenidas para una imagen de la secuencia.	201
Figura 7.6. (a) Detalle de la imagen de entrada sobre la que se han resaltado en color rojo los píxeles para los que se representa el valor de la profundidad. (b) Valor de la profundidad en los píxeles marcados en rojo en (a) en función del valor de la constante $\mu$ .	202
Figura 7.7. Valor máximo de la diferencia entre la profundidad obtenida para $\mu = 10^{-3}$ y para los siguientes valores de $\mu$ : $5 \cdot 10^{-3}$ , $10^{-2}$ , $5 \cdot 10^{-2}$ , $10^{-1}$ , $5 \cdot 10^{-1}$ , $10^0$ .	202
Figura 7.8. Errores cometidos en el cálculo de las velocidades lineal y angular calculados como la diferencia entre las velocidades estimadas por el algoritmo, y las medidas por los sensores odométricos del robot en $\Gamma_w$ . (a) Error cometido en el cálculo de la velocidad lineal. (b) Error cometido en el cálculo de la velocidad angular.	205
Figura 7.9. (a) Detalle de la imagen de entrada sobre la que se han resaltado en color rojo los píxeles para los que se representa el valor de la profundidad. (b) Valor de la profundidad en los píxeles marcados en rojo en (a) en función del valor de la constante $\mu$ .	208
Figura 7.10. Detalle de las curvas obtenidas para una imagen de la secuencia utilizando diferentes valores de la constante $\lambda$ .	210
Figura 7.11. Número de píxeles contenidos por la curva que define el contorno del robot móvil en función del valor de la constante $\mu$ , para 5 de las 100 imágenes de la secuencia de prueba. (a) Número de píxeles por la curva obtenida para la imagen de la cámara 1. (b) Número de píxeles en la curva	

obtenida para la imagen de la cámara 2. (c) Número de píxeles en la curva obtenida para la imagen de la cámara 3. (d) Número de píxeles en la curva obtenida para la imagen de la cámara 4. ....	216
Figura 7.12. Detalle de las curvas iniciales de una de las imágenes de la secuencia (a) Vista desde la cámara 1 (b) Vista desde la cámara 3. ....	217
Figura 7.13. (a) Número de píxeles en que se diferencia la curva inicial de la curva obtenida tras la minimización de energía para cada una de las cuatro cámaras disponibles. (b) Evolución media de las curvas por cada iteración (calculada dividiendo el número de píxeles en que difieren la curva inicial y la final entre el número medio de iteraciones). ....	223
Figura 7.14. Detalle de las curvas obtenidas para una imagen de la secuencia utilizando diferentes valores de la constante $\lambda$ . (a) Detalle de las curvas en la imagen captada por la cámara 1. (b) Detalle de las curvas en la imagen captada por la cámara 3. ....	223
Figura 8.1. Diagrama de bloques general del XPFCP propuesto en [Marron 2008]. Las etapas correspondientes al filtro de partículas extendido se muestran en color azul, mientras que las etapas de clasificación de medidas y de partículas se han representado en color naranja. ....	234
Figura 8.2. Diagrama de bloques general de la solución propuesta para la segmentación de movimiento y la estimación de la posición 3D de un número variable de robots móviles. ....	238
Figura 8.3. (a) Imágenes de entrada pertenecientes a la secuencia de pruebas vistas por tres cámaras fijas en el ISPACE-UAH. (b) Conjunto de puntos obtenidos tras la minimización de la función objetivo y la identificación de los robots móviles contenidos en la escena mostrada en las imágenes anteriores. ....	240
Figura 8.4. Representación gráfica del peso normalizado asociado a cada una de las partículas en el paso de corrección del XPF. Las coordenadas X e Y coinciden con las coordenadas de la partícula en $\Gamma_w$ . ....	242
Figura 8.5. Representación sobre el plano $X_w Y_w$ del resultado de la clasificación del conjunto de partículas mostrado en la Figura 8.4. ....	243
Figura 8.6. Cubos obtenidos a partir de los grupos de partículas a la salida del XPFCP. El interior de estos cubos define la zona donde existe una mayor probabilidad de encontrar robots móviles en la siguiente imagen de la secuencia. ....	244
Figura 8.7. Resultado de la proyección de los cubos definidos alrededor de cada grupo de partículas sobre el plano imagen de cada una de las tres cámaras. ....	245
Figura 8.8. Diagrama de bloques general del proceso de obtención del VH3D, utilizando una máscara, a partir de las curvas iniciales que incorpora la información obtenida del XPFCP en la imagen anterior. ....	246
Figura 9.1. Dos ejemplos (a) y (b) de escenas pertenecientes a la secuencia 1 adquiridas por cuatro cámaras en el ISPACE-UAH, de las cuales solamente se representa la información captada por tres cámaras. ....	252
Figura 9.2. Escena perteneciente a la secuencia 2 adquirida por tres cámaras en el ISPACE-UAH. ....	253
Figura 9.3. Imágenes pertenecientes a la secuencia 3 adquiridas por cuatro cámaras en el ISPACE-UAH. Vista desde tres de las cuatro cámaras de dos escenas en las que aparece un robot móvil. Las dos escenas han sido adquiridas en diferentes condiciones de iluminación. ....	254
Figura 9.4. Imágenes pertenecientes a la secuencia 4 adquiridas por cuatro cámaras en el ISPACE-UAH. Vista desde tres de las cuatro cámaras de dos escenas diferentes. (a) Imagen en la que aparece un robot móvil y una persona. (b) Imagen en la que únicamente aparece un robot móvil. ....	254
Figura 9.5. Imágenes pertenecientes a la secuencia 5 adquiridas por cuatro cámaras en el ISPACE-UAH. Vista desde tres de las cuatro cámaras de dos escenas diferentes. (a) Imagen en la que únicamente	

aparece el robot móvil (b) Imagen en la que se muestra tanto un robot como una persona que coloca diferentes obstáculos en el área de movimiento del robot. ....	255
Figura 9.6. (a) Tiempo de cómputo consumido en la inicialización de variables y la minimización de la función objetivo de 100 imágenes de la secuencia 1 (Figura 9.1). El color rojo se corresponde con la inicialización utilizando circunferencias y en verde utilizando la técnica GPCA. (b) Detalle del tiempo de cómputo para las imágenes entre la 150 y la 170. (c) Detalle del número de robots detectados en cada par de imágenes de entrada de la secuencia 1. ....	258
Figura 9.7. (a) Tiempo consumido en la inicialización de variables y la minimización de la función objetivo de 100 imágenes de la secuencia 2. En rojo se muestran los resultados inicializando las curvas de contorno con circunferencias y en verde utilizando GPCA. (b) Número de robots detectados en cada par de imágenes de entrada de la secuencia 1. ....	259
Figura 9.8. Detalle de las curvas iniciales (en rojo) y el resultado de la segmentación de movimiento (en verde) para la imagen 85 de la secuencia 1 (a) Al inicializar las curvas usando circunferencias (resultado obtenido tras 59 iteraciones). (b) Al inicializar las curvas mediante GPCA (resultado obtenido tras 2 iteraciones). Gran parte de la curva inicial no se puede ver porque el resultado de la segmentación se superpone a ella. ....	260
Figura 9.9. Resultado de la segmentación de movimiento de cuatro imágenes pertenecientes a la secuencia 3 (Figura 9.3) que han sido adquiridas en diferentes condiciones de iluminación. ....	261
Figura 9.10. Error cuadrático cometido en el posicionamiento de los robots móviles (parte superior) y tiempo de procesamiento (parte inferior) de cada par de imágenes de la secuencia 1. ....	262
Figura 9.11. (a) Error cuadrático cometido en el posicionamiento de los robots móviles (b) Tiempo de procesamiento de cada par de imágenes de la secuencia 2. ....	263
Figura 9.12. Trayectoria de los robots móviles a lo largo de las imágenes de la secuencia 2. Resultado de la proyección sobre el plano XY en $\Gamma_w$ . ....	264
Figura 9.13. Tiempo de procesamiento (gráfica superior) y error de posición (gráfica inferior) para 100 imágenes pertenecientes a la secuencia 1 (Figura 9.1). En rojo se muestran los resultados para el caso de la F.O para una sola cámara y en verde para la F.O para múltiples cámaras. En el caso del error de posición, las gráficas para ambas F.O. coinciden. ....	266
Figura 9.14. Tiempo de procesamiento (gráfica superior) y error de posición (gráfica inferior) para 100 imágenes pertenecientes a la secuencia 2 (Figura 9.2) obtenidas a partir de las imágenes adquiridas por una única cámara. En el caso del error de posición, las gráficas obtenidas para ambas funciones objetivo se superponen. ....	267
Figura 9.15. Tiempo de procesamiento (gráfica superior) y error de posición (gráfica inferior) para 100 imágenes pertenecientes a la secuencia 1 (Figura 9.1) obtenidas a partir de las imágenes adquiridas por dos (columna izquierda) y tres (columna derecha) cámaras. En el caso del error de posición, las gráficas para ambas funciones objetivo coinciden. ....	267
Figura 9.16. Tiempo de procesamiento (gráficas superiores) y error de posición (gráficas inferiores) para 100 imágenes pertenecientes a la secuencia 2 (Figura 9.2) obtenidos a partir de las imágenes adquiridas por dos (columna izquierda) y tres (columna derecha) cámaras. ....	268
Figura 9.17. Tiempo de cómputo (gráfica superior) y error de posición (gráfica inferior) calculados para 100 imágenes pertenecientes a la secuencia 1 (Figura 9.1). ....	271
Figura 9.18. Tiempo de cómputo (gráfica superior) y error de posición (gráficas intermedia e inferior) calculados para 100 imágenes pertenecientes a la secuencia 2 (Figura 9.2). ....	271
Figura 9.19. Valor medio del tiempo consumido por el XPFCP, obtenido para 100 imágenes pertenecientes a las secuencia 1 (Figura 9.1) y 2 (Figura 9.2). ....	273

Figura 9.20. Tiempo medio consumido para la segmentación de movimiento y la estimación de la posición 3D para 100 imágenes pertenecientes a las secuencias 1 y 2 en función del número de partículas consideradas en el XPFCP.....	273
Figura 9.21. Error de posición medio calculado para 100 imágenes pertenecientes a la secuencia 1 (gráfica superior) y a la secuencia 2 (gráfica inferior) en función del número de partículas consideradas en el XPFCP. ....	274
Figura 9.22. Curvas obtenidas de la segmentación de movimiento (columna izquierda) y resultado de la reconstrucción 3D mediante VH3D (columna derecha) para dos imágenes pertenecientes a la secuencia 1 (Figura 9.1). Además, se muestran la trayectoria real y la estimada.....	276
Figura 9.23. Curvas obtenidas de la segmentación de movimiento (columna izquierda) resultado de la reconstrucción 3D mediante VH3D (columna derecha) para tres imágenes pertenecientes a la secuencia 2 (Figura 9.2). En todas las imágenes se representa también la trayectoria estimada..	277
Figura 9.24. Curvas de la segmentación de movimiento (columna izquierda) resultado de la reconstrucción 3D mediante VH3D (columna derecha) obtenida a partir de los contornos de la segmentación para dos imágenes pertenecientes a la secuencia 3 (Figura 9.3).....	278
Figura 9.25. Curvas de la segmentación de movimiento (columna izquierda) resultado de la reconstrucción 3D mediante VH3D (columna derecha) obtenida a partir de los contornos de la segmentación para dos imágenes pertenecientes a la secuencia 4 (Figura 9.4).....	279
Figura 9.26. Curvas de segmentación de movimiento (columna izquierda) resultado de la reconstrucción 3D mediante VH3D (columna derecha) obtenida a partir de los contornos de la segmentación para dos imágenes pertenecientes a la secuencia 5 (Figura 9.5). ....	280
Figura 9.27. Tiempo de procesamiento de cada par de imágenes pertenecientes a cada una de las secuencias de test. ....	281
Figura 9.28. Valor medio del tiempo de cómputo consumido para la segmentación de movimiento y el posicionamiento 3D de múltiples robots móviles en cada una de las secuencias de test. ....	282
Figura 9.29. Error de posición calculado utilizando la ecuación (9.1) para cuatro secuencias de prueba. (a) Secuencia 1 (Figura 9.1). (b) Secuencia 2 (Figura 9.2). (c) Secuencia 3 (Figura 9.3). (d) Secuencia 4 (Figura 9.4). ....	283
Figura 9.30. Valor medio del error de posición calculado utilizando la ecuación (9.1) para cada uno de los robots incluidos en las secuencias de test 1 (Figura 9.1), 2 (Figura 9.2), 3 (Figura 9.3) y 4 (Figura 9.4). ....	283
Figura 9.31. Trayectoria estimada representada sobre el plano imagen de la cámara 1 de las secuencias de prueba. (a) Secuencia 1 (Figura 9.1). (b) Secuencia 2 (Figura 9.2). (c) Secuencia 3 (Figura 9.3). (d) Secuencia 4 (Figura 9.4).....	284
Figura 9.32. Trayectoria estimada a lo largo de cada secuencia y reconstrucción 3D de la imagen 100 de las secuencias de prueba. (a) Secuencia 1 (Figura 9.1). (b) Secuencia 2 (Figura 9.2). (c) Secuencia 3 (Figura 9.3). (d) Secuencia 4 (Figura 9.4).....	285
Figura 9.33. Trayectoria estimada a partir de 500 imágenes pertenecientes a la Secuencia 2 (Figura 9.2). Proyección sobre el plano (XY) en $\Gamma_w$ . ....	286
Figura 10.1. Diagrama de bloques general del algoritmo de segmentación de movimiento rígido y obtención de la posición 3D a partir de las imágenes captadas por $n$ cámaras ubicadas en posiciones fijas del espacio. ....	289





# Índice de Tablas

Tabla 4.1. Criterio de convergencia: valor máximo de variación de la variable entre una iteración y la siguiente para el que se considera que la variable converge. Se indican los diferentes grupos de variables implicados en la función objetivo de la ecuación (4.14).....	65
Tabla 4.2. Criterio de convergencia: valor máximo de variación de la variable entre una iteración y la siguiente para el que se considera que la variable converge para cada uno de los grupos de variables implicados en la función objetivo de la ecuación (4.66).....	83
Tabla 5.1. Valores de los errores de recuperación en función del número de imágenes y autovectores considerados en el cálculo del modelo de fondo usando GPCA .....	92
Tabla 5.2. Número de elementos de las matrices implicadas en la obtención del modelo de fondo utilizando las técnicas PCA y GPCA. ....	96
Tabla 5.3. Tiempo de cómputo consumido para el cálculo de las matrices de transformación de PCA y GPCA en función del número de imágenes y de autovectores considerados para la obtención del modelo de fondo. ....	97
Tabla 5.4. Error de recuperación medio en función del número de imágenes y autovectores considerados para la obtención del modelo de fondo. El error de recuperación se ha obtenido como la media del valor de recuperación medio obtenido para 15 imágenes de fondo. ....	98
Tabla 5.5. Valor de la constante $K$ óptima, elegida manualmente para el cálculo del umbral utilizando la ecuación (5.14) y valor del umbral para un conjunto de imágenes de prueba pertenecientes a las secuencias de las imágenes representadas en la Figura 5.11 (a) y (b). ....	112
Tabla 5.6. Porcentaje de píxeles segmentados correctamente, falsos positivos y píxeles pertenecientes a los robots sin segmentar para las imágenes de la secuencia 1 (Figura 5.20 (a)) y la secuencia 2 (Figura 5.20 (b)). ....	116
Tabla 5.7. Valor inicial del umbral en función del tamaño de la ventana utilizada para el cálculo del modelo de fondo y número de píxeles segmentados correctamente (en porcentaje).....	120
Tabla 5.8. Valor inicial del umbral en función del tamaño de la ventana utilizada para el cálculo del modelo de fondo actualizando el valor de la constante $K$ en función del tamaño de la ventana, y porcentaje de píxeles segmentados de forma correcta. ....	124
Tabla 5.9. Número de píxeles segmentados de forma correcta tras la umbralización del error de recuperación aplicando el umbral proporcional (5.14) con $K=10$ , respecto al caso de la segmentación manual, cuando la secuencia de imágenes de entrada se ha modificado incrementando o reduciendo entre -100 y 100 el nivel de gris de todos los píxeles uniformemente. ....	131
Tabla 5.10. Magnitud del rango de valores en que se puede incrementar la intensidad de las imágenes de entrada sin que se deterioren las curvas iniciales obtenidas en función del valor de las constantes $\alpha$ y $\beta$ consideradas en el cálculo del umbral (5.18).....	135

Tabla 5.11. Número de píxeles (en porcentaje) segmentados de forma correcta tras incrementar la intensidad de todos los píxeles de la imagen de entrada en un valor entre -50 y +50. ....	136
Tabla 5.12. Número de píxeles (en porcentaje) segmentados de forma correcta tras incrementar la intensidad de cada píxel de la imagen de entrada en un valor aleatorio entre -50 y +50 utilizando un umbral proporcional con $K$ constante (5.14) y un umbral adaptativo (5.18). ....	139
Tabla 5.13. Número de píxeles segmentados de forma correcta tras la umbralización del error de recuperación aplicando el umbral adaptativo definido en la ecuación (5.18) sobre las imágenes de entrada pertenecientes a las secuencias 3 y 4 (mostradas en la Figura 5.48) a las que se les ha modificado la intensidad media sumando un valor aleatorio entre -50 y 50 a cada píxel de cada imagen. ....	142
Tabla 5.14. Valores de los umbrales $U_g$ y $U_{inv}$ obtenidos de forma supervisada para cada una de las imágenes de prueba de las secuencias 1 (Figura 5.56) y 2 (Figura 5.57). ....	151
Tabla 5.15. Porcentaje de píxeles segmentados de forma correcta en función de los diferentes valores de los umbrales para la detección de sombras en la imagen $U_g$ y $U_{inv}$ considerados para un conjunto de imágenes de la secuencia 1 (a la que pertenecen la imágenes de la Figura 5.56). ....	153
Tabla 5.16. Porcentaje de píxeles segmentados de forma correcta en función de los diferentes valores de los umbrales para la detección de sombras en la imagen $U_g$ y $U_{inv}$ considerados para un conjunto de imágenes de la secuencia 2 (a la que pertenece la imagen de la Figura 5.57). ....	153
Tabla 6.1. Tiempo de cómputo medio (en segundos) consumido en el proceso de eliminación de sombras de las curvas iniciales. ....	172
Tabla 6.2. Valor de la distancia máxima entre los píxeles obtenidos tras proyectar un cubo de dimensiones 50x50x100 centímetros del sistema de referencia del mundo $\Gamma_w$ al plano imagen. ....	182
Tabla 6.3. Porcentaje de errores de fusión y duplicación de clases en función del factor de multiplicación utilizado para fijar el valor de la distancia límite. Resultados obtenidos utilizando la distancia euclídea (ecuación (6.3)), la distancia euclídea ponderada con $\tau=2/3$ (ecuación (6.5)) y la distancia sobre el eje $x$ (ecuación (6.4)). ....	183
Tabla 6.4. Tasa de error de duplicación de clases y unión de dos clases obtenidos a partir de una secuencia de 100 imágenes, en función del valor de $dMax$ . ....	188
Tabla 7.1. Resumen de las diferentes funciones para la preservación de bordes propuestas en [Sekkati & Mitiche 2006b] y [Aubert et al. 1999]. ....	195
Tabla 7.2. Rango de valores y tiempo de cómputo consumido en el cálculo del término dependiente del gradiente de la profundidad en la ecuación de descenso por el gradiente, para las diferentes funciones de preservación de bordes ( $g(s)$ ) consideradas. Valores medios obtenidos para 100 imágenes de prueba capturadas por 4 cámaras. ....	197
Tabla 7.3. Orden de magnitud de los valores del segundo término de la función objetivo (ecuación (7.1)) para las imágenes de prueba y de la constante $\mu$ , para que los dos términos de dicha ecuación tomen valores en el mismo orden de magnitud. ....	198
Tabla 7.4. Valor medio del error cometido en la estimación de la velocidad lineal y número medio de iteraciones necesarias hasta la convergencia del algoritmo para las 100 imágenes de la secuencia de prueba en función del valor de la constante $\mu$ , para valores de $\mu$ entre $10^{-3}$ y 1. ....	205
Tabla 7.5. Valor medio del error cometido en la estimación de la velocidad lineal y número medio de iteraciones necesarias hasta la convergencia del algoritmo, para las 100 imágenes de la secuencia de prueba para valores de $\mu$ entre $5 \cdot 10^{-2}$ y 1. ....	206
Tabla 7.6. Valor máximo de la diferencia entre la profundidad, en mm, calculada con $\lambda = 20000$ , y la obtenida con $\lambda = [2, 20, 200, 2000]$ . ....	208

Tabla 7.7. Número de iteraciones, diferencia entre la curva inicial y el resultado de la segmentación, factor de evolución y evolución media de la curva por cada iteración, para $\lambda = [2 \cdot 10^{-1}, 2 \cdot 10^0, 2 \cdot 10^1, 2 \cdot 10^2, 2 \cdot 10^3 \text{ y } 2 \cdot 10^4]$ .....	209
Tabla 7.8. Error cuadrático medio, en función del valor de la constante $\lambda$ , cometido en la estimación de la velocidad lineal y angular, y número medio de iteraciones necesarias hasta la convergencia del algoritmo para las 100 imágenes de la secuencia de prueba (a la que pertenecen las imágenes de la Figura 7.1) para valores de $\lambda$ entre $2 \cdot 10^{-1}$ y $2 \cdot 10^4$ . .....	211
Tabla 7.9. Valor medio del error cometido en la estimación de la velocidad lineal y número medio de iteraciones necesarias hasta la convergencia del algoritmo para las 100 imágenes de la secuencia de prueba en función del valor de la constante $\lambda$ , para valores de $\lambda$ entre 100 y 2000. ....	212
Tabla 7.10. Número medio de iteraciones, y error cuadrático medio de la velocidad lineal (en mm/s) en función del valor de las constantes $\lambda$ y $\mu$ . Los resultados se obtienen para 25 imágenes de las secuencias 1 (Figura 7.1), 2 (Figura 7.2(a)) y 3 (Figura 7.2 (b)). .....	213
Tabla 7.11. Valor medio de la profundidad de los puntos contenidos en la curva que define el contorno del robot móvil para una imagen de la secuencia 1 (a la que pertenecen las imágenes de la Figura 7.1) en función del valor de la constante $\mu$ . .....	218
Tabla 7.12. Valor medio del error cometido en la estimación de la velocidad lineal y número medio de iteraciones necesarias hasta la convergencia del algoritmo para las 100 imágenes de la secuencia de prueba (a la que pertenece la imagen de la Figura 7.1) en función del valor de la constante $\mu$ , para valores de $\mu$ entre $10^{-3}$ y 1. ....	219
Tabla 7.13. Valor medio del error cometido en la estimación de la velocidad lineal y número medio de iteraciones necesarias hasta la convergencia del algoritmo para las 100 imágenes de la secuencia de prueba (a la que pertenecen las imágenes de la Figura 7.1) en función del valor de la constante $\mu$ , para valores de $\mu$ entre $3 \cdot 10^{-3}$ y $10^{-1}$ .....	220
Tabla 7.14. Valor medio de la profundidad, en función del valor de la constante $\lambda$ , de los puntos contenidos en la curva que define el contorno del robot móvil para una imagen de la secuencia 1. ....	221
Tabla 7.15. Número de iteraciones, diferencia entre la curva inicial y el resultado de la segmentación y factor de evolución de la curva para $\lambda = [2 \cdot 10^{-1}, 2 \cdot 10^0, 2 \cdot 10^1, 2 \cdot 10^2, 2 \cdot 10^3 \text{ y } 2 \cdot 10^4]$ . ....	222
Tabla 7.16. Error cuadrático medio cometido en la estimación de la velocidad lineal y angular, y número medio de iteraciones necesarias hasta la convergencia del algoritmo para las 100 imágenes de la secuencia de prueba, en función del valor de la constante $\lambda$ , para valores de $\lambda$ entre $2 \cdot 10^{-1}$ y $2 \cdot 10^4$ . ....	224
Tabla 7.17. Valores medios de los errores cometido en la estimación de la velocidad lineal y angular y número medio de iteraciones necesarias hasta la convergencia del algoritmo para las 100 imágenes de la secuencia de prueba, en función del valor de la constante $\lambda$ , para valores de $\lambda$ entre $2 \cdot 10^3$ y $2 \cdot 10^4$ . ....	225
Tabla 7.18. Error cuadrático medio de la velocidad lineal (en milímetros/segundo) en función del valor de las constantes $\lambda$ y $\mu$ . Los resultados se obtienen para 25 imágenes de las secuencias 1, 2 y 3 (a las que pertenecen las imágenes de la Figura 7.1 y la Figura 7.2). ....	226
Tabla 7.19. Número medio de iteraciones en función del valor de las constantes $\lambda$ y $\mu$ . Los resultados se obtienen para 25 imágenes de las secuencias 1 (Figura 7.1), 2 (Figura 7.2(a)) y 3 (Figura 7.2(b)). ....	227
Tabla 9.1. Tiempo de cómputo medio (en segundos) consumido en las etapas de inicialización y minimización de la función objetivo y número medio de iteraciones necesarias para 100 imágenes de las secuencias 1 y 2, en función de las curvas iniciales utilizadas. ....	259

Tabla 9.2. Error de posicionamiento medio (en milímetros) y tiempo de cómputo medio (en segundos) consumido en las etapas de inicialización y minimización de la función objetivo obtenidas para 100 imágenes de las secuencias 1 y 2, en función del método de inicialización de la profundidad utilizado. ....	264
Tabla 9.3. Valor medio del tiempo de procesamiento y del error cometido en la estimación de la posición 3D, obtenido para 100 imágenes pertenecientes a la secuencia 1, en función del número de cámaras y la función objetivo utilizada.....	269
Tabla 9.4. Valor medio del tiempo de procesamiento y del error cometido en la estimación de la posición 3D, obtenido para 100 imágenes pertenecientes a la secuencia 2, en función del número de cámaras y de la función objetivo utilizada. ....	269
Tabla 9.5. Valor medio del tiempo de procesamiento y del error de posicionamiento 3D calculado para 100 imágenes pertenecientes a las secuencias 1 y 2.....	272
Tabla 9.6. Valor medio del tiempo de procesamiento y del error de posición en función del número de partículas del XPFCP, calculados para 100 imágenes pertenecientes a las secuencias 1 y 2.....	275
Tabla 9.7. Valor medio del tiempo de procesamiento de las diferentes etapas de la solución propuesta y error medio de posición. Resultados obtenidos para 100 imágenes pertenecientes a cada una de las secuencias de test. ....	284

Capítulo 1

# **Introducción**



La evolución de la electrónica y las comunicaciones durante las últimas décadas ha permitido que las tecnologías de la información se instauren en los diferentes ámbitos de la vida cotidiana de las personas. En esta línea, una parte importante de los esfuerzos de los investigadores en los últimos años se ha centrado en la creación de “entornos inteligentes” en los que los usuarios puedan interactuar de manera natural con los diferentes sistemas y servicios computacionales que les faciliten la realización de sus tareas diarias.

Entre los antecedentes más significativos de los espacios inteligentes (también conocidos como ambientes inteligentes, o *smart rooms*) se encuentra el concepto de computación ubicua, propuesto inicialmente por Weiser en 1991 en su obra "*The Computer for the Twenty-First Century*", y que engloba los conceptos de inmersión y transparencia de la tecnología para el usuario. En este sentido, un espacio inteligente cuenta con una red de sensores que le permiten adquirir información del entorno así como un conjunto de actuadores para la interacción con los usuarios, siendo tanto la red de sensores, como la inteligencia que controla a los diferentes agentes, imperceptible para el usuario.

Las aplicaciones de los espacios inteligentes son numerosas y se encuentran en auge en la actualidad, abarcando diferentes ámbitos tales como la robótica de servicios, tareas de vigilancia automática, ayudas a personas con discapacidad o los servicios domóticos.

En la presente tesis se denomina espacio inteligente a un área física dotada de sensores. Dichos sensores son controlados por un sistema de supervisión dotado de capacidad de análisis y toma de decisiones. La inteligencia del sistema está constituida por uno o varios nodos interconectados entre sí, que realizan la captura y análisis de la información del entorno adquirida por los sensores. El espacio inteligente dispone además de un conjunto de actuadores que permiten la interacción con los usuarios. Entre estos actuadores destacan los robots móviles controlados por el sistema de supervisión.

Una de las tareas esenciales en los espacios inteligentes, cuando la actuación se realiza mediante robots móviles, es la localización e identificación de dichos robots, así como de los posibles obstáculos (fijos o móviles) y los usuarios dentro del espacio inteligente, con suficiente precisión. La presente tesis doctoral surge con el objetivo de realizar contribuciones a esas tareas, en concreto, se pretende desarrollar un sistema que permita realizar la segmentación e identificación de los robots móviles (agentes

controlados) en el espacio inteligente a partir de las imágenes adquiridas por un conjunto de cámaras ubicadas en posiciones fijas del espacio.

El objetivo perseguido por esta tesis conlleva una serie de tareas que deben llevarse a cabo dentro de la línea de investigación planteada. Dichas tareas se exponen en los siguientes apartados de este capítulo, junto con el esquema general de la solución propuesta en esta tesis y los contenidos de los diferentes capítulos que componen esta memoria.

## **1.1. Objetivos generales de la tesis**

El objetivo principal de esta tesis es la segmentación, identificación y posicionamiento 3D de robots móviles en un espacio inteligente, a partir de la información adquirida por una o varias cámaras sincronizadas y calibradas, ubicadas en posiciones fijas del entorno en que se desplazan los robots. Cabe destacar que la solución propuesta permite también la localización de otros elementos móviles presentes en el entorno (como pueden ser los usuarios, o los posibles obstáculos móviles), sin embargo, el trabajo desarrollado se centra únicamente en los agentes controlados que se mueven por el espacio inteligente. Por este motivo, en esta memoria haremos referencia únicamente a robots móviles.

En este sentido, la tesis se enmarca dentro del área de los “espacios inteligentes”, entendiendo este término como un área física dotada de sensores que le permiten percibir y entender qué está sucediendo. Dichos sensores son controlados por un sistema de supervisión que toma decisiones en función de la información disponible. El espacio inteligente cuenta además con actuadores que le permiten interactuar con el entorno y los usuarios del mismo, destacando, entre estos actuadores, los robots móviles controlados por el sistema (agentes controlados).

En el caso concreto de esta tesis, el objetivo planteado es la localización de estos agentes controlados, dotando al espacio por el que se desplazan de un conjunto de cámaras calibradas dispuestas estratégicamente, de forma que cubran el área de movimiento de los robots. Todas las cámaras se encuentran conectadas a una red de comunicaciones que permite que la adquisición de las mismas se realice de forma sincronizada, siendo controlada por el sistema de supervisión del espacio inteligente.



Este sistema de supervisión controla además los robots móviles, a los que está conectado mediante una red inalámbrica. En la Figura 1.1 se muestra un esquema general en el que se presentan los principales elementos del espacio inteligente considerado en esta tesis.

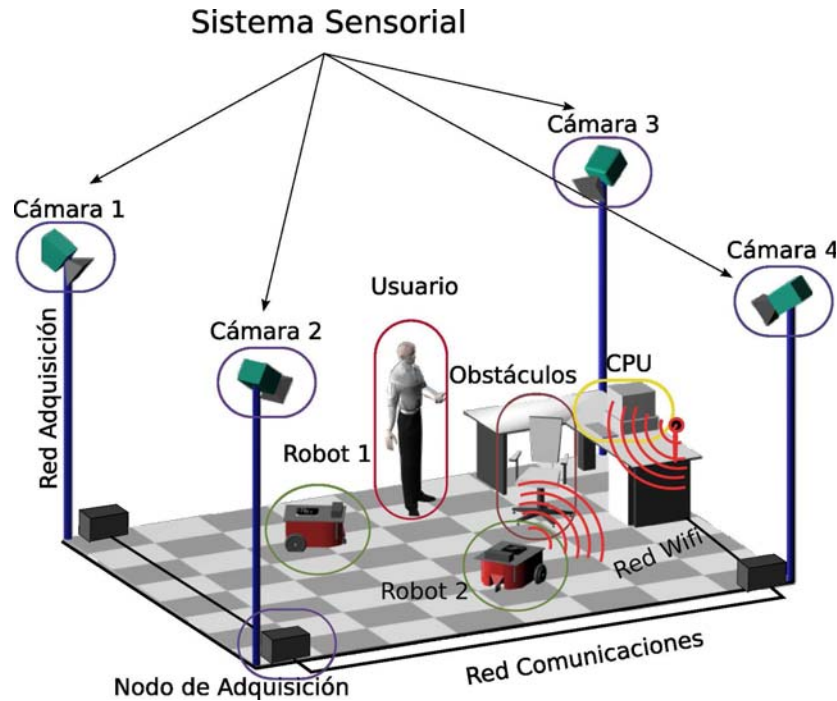


Figura 1.1. Estructura general del Espacio Inteligente en el que se representan los diferentes elementos del mismo.

En el entorno de trabajo, pueden existir tanto agentes controlados (robots móviles) por el Espacio Inteligente que permiten al sistema interactuar con el entorno, como agentes no controlados, entre los que se incluyen tanto los posibles usuarios, como los obstáculos (fijos o móviles) existentes en el entorno en que se desplazan los robots móviles.

Entre las características del sistema de segmentación y posicionamiento de robots móviles desarrollado en esta tesis destacan las siguientes:

- Permite la detección y segmentación, así como la obtención de la posición 3D de un número variable de robots móviles en un espacio inteligente.
- La segmentación y posición 3D de los robots pueden obtenerse incluso si sólo se cuenta con las imágenes adquiridas por una cámara (bajo el supuesto de que los robots se desplazan en un plano). En caso de que el número de cámaras sea mayor, la información adquirida por todas ellas se incluye en el algoritmo de segmentación y posicionamiento con objeto de incrementar la robustez del

sistema frente a cambios en la iluminación, sombras u oclusiones, así como para mejorar la precisión de los resultados de segmentación y posicionamiento.

- No se precisa información a priori acerca del número de objetos móviles presentes en la escena, ni de las características físicas o la estructura tridimensional de los mismos. Además, no es necesario incorporar marcas artificiales o balizas a bordo de los robots móviles para su identificación.
- Los robots móviles a segmentar disponen de sensores odométricos a bordo cuyas medidas se transmiten al espacio inteligente vía radio. Tanto la información medida por los sensores odométricos, como los comandos de guiado enviados a los robots móviles están disponibles en el sistema de supervisión del espacio inteligente, pudiendo incorporarse, en caso de que sea necesario, para mejorar las características del algoritmo de segmentación y posicionamiento 3D.
- Finalmente, el sistema sensorial utilizado está compuesto por un conjunto de cámaras calibradas ubicadas en posiciones fijas del Espacio Inteligente. Estas cámaras tienen la óptica apropiada y están estratégicamente ubicadas en puntos alejados (con la distancia entre las cámaras y el espacio en el que se mueven los robots grande) de forma que todos los puntos del espacio de movimiento de los robots son vistos por el mayor número de cámaras posible (como mínimo dos).

## **1.2. Esquema general del sistema propuesto**

Para alcanzar los objetivos propuestos en esta tesis, bajo las condiciones descritas en el apartado anterior, es necesario llevar a cabo un conjunto de tareas que permitan resolver los diferentes problemas que se presentan dentro de la línea de investigación planteada. El objetivo final de todas las tareas es el desarrollo de un sistema completo que permita segmentar los robots móviles presentes en el espacio inteligente y estimar la posición 3D de los mismos con suficiente precisión. Dicha solución debe ser, además, robusta frente a la presencia de cambios en la iluminación, sombras u oclusiones en las imágenes adquiridas por las diferentes cámaras del espacio inteligente.

En la Figura 1.2 se muestra, en forma de diagrama de bloques, un esquema general de la solución propuesta en esta tesis. En esta figura pueden observarse las diferentes etapas de las que consta el algoritmo de segmentación y posicionamiento 3D de robots móviles propuesto, representándose en color azul las etapas desarrolladas en la presente tesis doctoral.

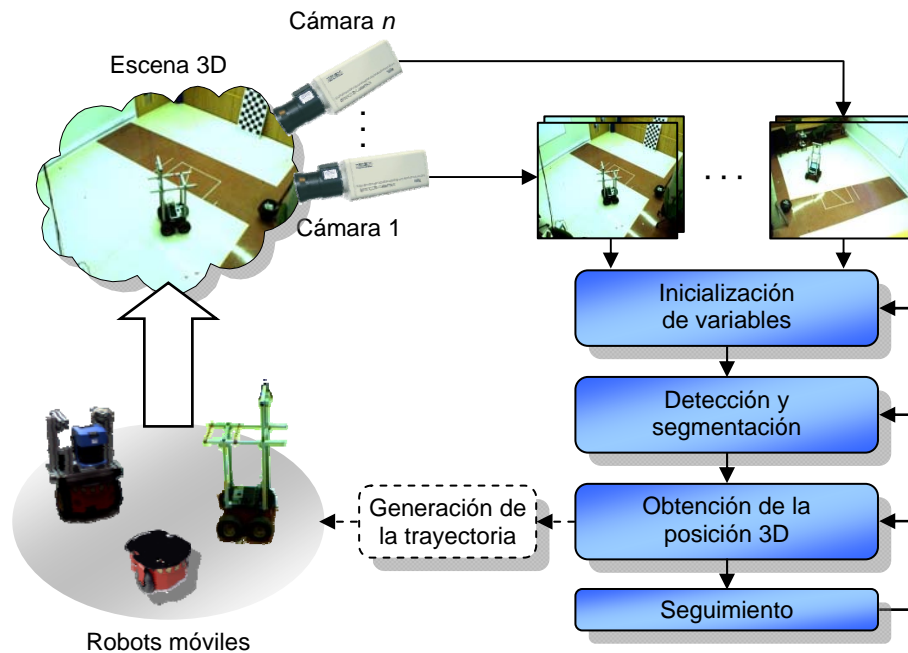


Figura 1.2. Esquema general de la solución propuesta en esta tesis. Los bloques representados en color azul representan las diferentes tareas desarrolladas para la fase de segmentación y obtención del posicionamiento.

La ejecución del algoritmo cuyas etapas se representan en la Figura 1.2 comienza con la inicialización de las diferentes variables implicadas en dicho proceso. Tal como se verá más adelante, previa a la segmentación de movimiento, es necesario obtener un conjunto de curvas iniciales, que dividen cada una de las imágenes en  $N$  regiones (asociadas a los  $N-1$  robots móviles y al fondo), así como la profundidad de cada uno de los puntos de las imágenes consideradas. Este proceso de inicialización tiene gran importancia debido a que, como se demuestra en la tesis, tanto el tiempo de cómputo, como la precisión de los resultados de detección y segmentación dependen fuertemente de los valores iniciales de las variables implicadas.

Tras la inicialización de variables, se lleva a cabo la detección y segmentación de los robots móviles. En esta etapa también puede llevarse a cabo la diferenciación entre los robots y los obstáculos móviles presentes en la escena utilizando para ello la información de los comandos de guiado a los robots y las medidas de los sensores odométricos a bordo de los mismos. Una vez identificados los robots móviles, se

procede a la obtención de su posición 3D. Cabe destacar que el algoritmo propuesto en esta tesis permite realizar la segmentación y el posicionamiento de los robots móviles de forma concurrente, en lugar de secuencial, mediante la minimización de una función objetivo obtenida que depende de diferentes variables relacionadas con las velocidades y la posición 3D de los robots.

Conocida la posición 3D de los robots móviles, es posible generar las trayectorias que deben seguir y los correspondientes comandos de guiado. Las tareas relacionadas con la generación de trayectorias y los algoritmos de guiado de los robots no es objetivo de esta tesis.

Por otro lado, el resultado de las etapas de segmentación y posicionamiento 3D de los robots móviles se introduce como entrada del último bloque del algoritmo. En esta última etapa se realiza el seguimiento de los robots móviles mediante un filtro de partículas. La salida del módulo seguidor se realimenta a las diferentes etapas del algoritmo de segmentación y posicionamiento 3D con el objetivo de incrementar la robustez del mismo, así como reducir el tiempo consumido en la inicialización de variables, y facilitar el proceso de identificación de los robots móviles.

### **1.3. Estructura de la tesis**

A lo largo de los 10 capítulos de los que consta la memoria de esta tesis se incluye un estudio teórico exhaustivo de los diferentes conceptos implicados en la solución desarrollada, así como la validación experimental de las diferentes propuestas realizadas.

Para comenzar, en el capítulo 2 de la tesis se realiza un estudio acerca de los trabajos previos llevados a cabo por la comunidad científica en el área de interés. Se ha, prestando especial atención a las diferentes técnicas que permiten la segmentación y localización en condiciones similares a las planteadas en esta tesis, es decir, a partir de imágenes adquiridas por cámaras fijas en el entorno.

En el capítulo 3 se presenta en detalle el concepto de espacio inteligente, describiendo las características y los elementos del mismo. En este capítulo también se describe el espacio inteligente de la Universidad de Alcalá (ISPACE-UAH) en el que se han realizado todas las pruebas experimentales presentadas en la tesis.

El algoritmo de segmentación de movimiento y posicionamiento 3D basado en la minimización de una función objetivo se describe en el capítulo 4, donde se explican las diferentes funciones objetivo consideradas para el caso de una y varias cámaras, así como el proceso de minimización de dichas funciones.

Dada la importancia de la inicialización de los parámetros implicados en el algoritmo de segmentación y posicionamiento, en los capítulos 5 y 6 se presentan, respectivamente, la inicialización de las curvas que definen los contornos iniciales de los posibles robots móviles y de la profundidad para cada uno de los píxeles de las imágenes de entrada.

Los resultados obtenidos tras la minimización de la función objetivo también dependen fuertemente de la función de preservación de bordes elegida, así como de los valores de las constantes que ponderan la contribución de cada uno de los términos de dicha función objetivo. Estos parámetros se estudian a lo largo del capítulo 7, donde se explica el proceso seguido para la obtención los valores más adecuados para dichas constantes.

El algoritmo de seguimiento de los robots móviles se describe en el capítulo 8, en el que además se indican las diferentes ventajas derivadas de la realimentación de la salida del seguidor a las diferentes etapas del algoritmo de segmentación y posicionamiento.

En el capítulo 9 se muestran los resultados de las diferentes pruebas experimentales realizadas con el objetivo de validar la solución propuesta en esta tesis.

Finalmente, en el capítulo 10 se resume el sistema completo además de presentar las conclusiones finales de la tesis, junto con un resumen de las aportaciones realizadas en la misma, así como las líneas de trabajos futuros relacionadas con el trabajo desarrollado.



## Capítulo 2

# **Revisión de conocimientos**





## 2.1. Introducción

La obtención de la posición 3D a partir de la información adquirida por un sistema sensorial es una tarea fundamental en el campo de la robótica móvil para la que se han propuesto múltiples métodos. Estos métodos pueden clasificarse en dos grandes grupos en función de si los sensores se ubican a bordo de los robots [Kin 1993], [Luo et al. 2008], [Fu & Yang 2009], o en el entorno de movimiento de los mismos. Los métodos incluidos en el segundo grupo, entre los que se encuentran los espacios inteligentes, presentan diversas ventajas respecto al uso de sensores a bordo de los robots: en primer lugar, la complejidad de la electrónica a bordo de los robots puede reducirse (con las ventajas que ello conlleva), además, la navegación simultánea de múltiples robots en el mismo entorno no requiere incrementar la complejidad de la infraestructura. Por otro lado, la información adquirida por los sensores externos es más completa que la que puede obtener cada uno de los robots de forma individual, esto permite obtener información acerca de todos los robots, facilitando la cooperación entre ellos.

En la presente tesis se denomina espacio inteligente a un área física dotada de sensores, que son controlados por un sistema de supervisión con capacidad de análisis y toma de decisiones, así como de actuadores que le permiten interactuar con los usuarios. En este sentido, uno de los antecedentes más significativos de los espacios inteligentes (también conocidos como ambientes inteligentes, *smarts rooms*, *Intelligent Spaces* – *ISPACE*- en la literatura anglosajona) es el concepto de computación ubicua, propuesto inicialmente por Weiser [Weiser 1993], [Weiser 1999], y que supone la integración de la tecnología en el entorno, de forma que ésta sea transparente para el usuario. Siendo, uno de los primeros sistemas ubicuos, el implementado en el Laboratorio de Ciencias de la Computación del Xerox PARC (*Palo Alto Research Center*), dentro del proyecto *Ubicomp*, entre 1988 y 1994.

Desde los 90 hasta hoy, diversos grupos han desarrollado los conceptos de computación ubicua y espacios inteligentes.

Uno de los primeros proyectos relacionados con espacios inteligentes fue el desarrollado por el grupo de investigación VisMod (Vision and Modelling) del *Media Lab*. del MIT (Massachusetts Institute of Technology) bajo el nombre “*smart rooms*”

[Pentland 1997]. En este trabajo se propone una habitación equipada con una red de sensores formado por cámaras y micrófonos. Estos sensores permiten la localización de los usuarios, así como la identificación de los mismos a partir de su voz y apariencia. El sistema propuesto permite además el reconocimiento de algunos gestos y posturas de los usuarios que se encuentran dentro del ISPACE. En la Figura 2.1 se muestra una imagen perteneciente a un video de demostración de las capacidades de la *Smart Room* para la interacción de una persona con una mascota virtual. Este proyecto, así como el grupo de investigación VisMod se encuentran inactivos desde 2004.



Figura 2.1. Imagen perteneciente a un video en que se demuestran las capacidades de la “Smart Room” desarrollada por el grupo VisMod del MIT para la interacción entre una persona y una mascota virtual. (Imagen extraída de la página web del proyecto: <http://vismod.media.mit.edu/vismod/demos/smartroom/>)

También en el MIT, en el Laboratorio de Ciencias de la Computación e Inteligencia Artificial, se desarrolló un sistema conocido como “*Intelligent Room*” [Brooks 1997], [Coen 1998]. En este entorno surge el proyecto Oxygen [Look & Shrobe 2004], [Look et al. 2005], [Look & Shrobe 2007] en el que se utilizan sensores de audio y vídeo con objeto de permitir la interacción natural (mediante voz y gestos) de los usuarios con el entorno, de forma que sea posible la interpretación de sus intenciones. En la Figura 2.2 se muestra un diagrama general en el que se incluyen los diferentes elementos implicados en el proyecto Oxygen.

Otro proyecto importante relacionado con la interacción hombre-máquina en espacios inteligentes es el proyecto CHIL (Computers in the Human Interaction Loop). Este proyecto, coordinado por el laboratorio de Sistemas Interactivos de la Universidad Karlsruhe (TH) en Alemania, y en el que participan diversos centros de investigación de Europa y Estados Unidos, tiene como objetivo la creación de entornos en los que el sistema es capaz de modelar e interpretar el comportamiento de los usuarios, así como

prestarles asistencia de forma transparente. [Zhang et al. 2006], [Potamianos et al. 2006], [Zhang et al. 2007], [Stiefelhagen et al. 2008].

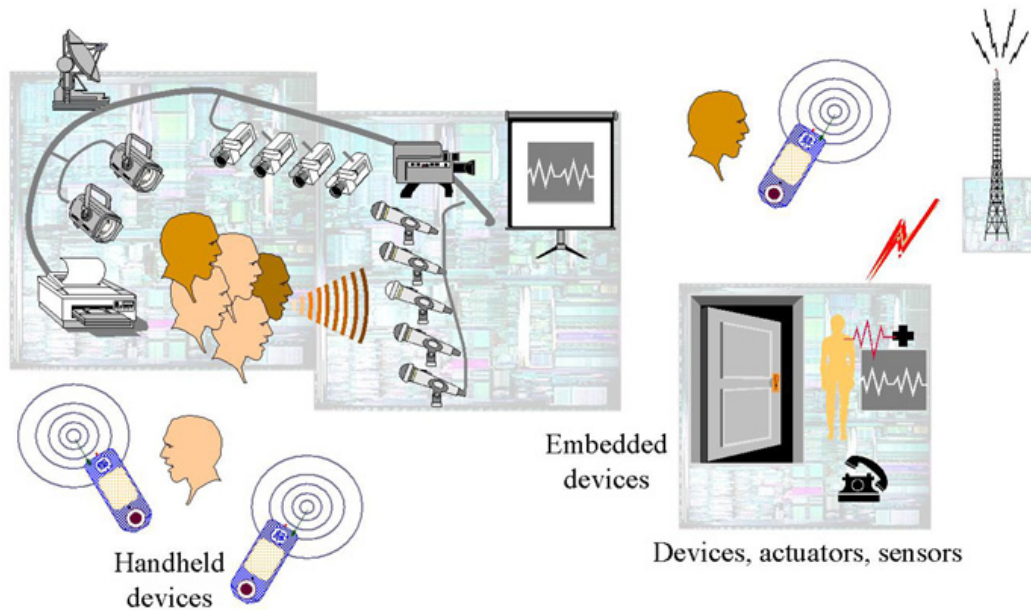


Figura 2.2. Esquema general del proyecto “Oxygen” desarrollado en el laboratorio de Ciencias de la Computación e Inteligencia Artificial del MIT.

Por otro lado, son varios los grupos que trabajan en el desarrollo de salas de reuniones inteligentes (*Smart meeting rooms*) en las que, mediante la combinación de cámaras y arrays de micrófonos, se realizan tareas como la localización, identificación y seguimiento de la persona que habla en cada momento. En esta línea se encuentran los trabajos realizados en el grupo SAIL (*Signal Analysis and Interpretation Laboratory*) de la Universidad Southern en California [Busso et al. 2005], [Busso et al. 2007], [Rozgic et al. 2007]. La Figura 2.3 muestra dos imágenes pertenecientes a la “smart meeting room” desarrollada en este centro.



(a)



(b)

Figura 2.3. Imágenes de la “*Smart meeting room*” desarrollada por el grupo SAIL de la USC. (a) Imagen general de (b) Detalle del array de micrófonos y la cámara omnidireccional

En el instituto de investigación IDIAP en Suiza también se desarrollan trabajos para la identificación y seguimiento de personas [Gatica-Perez et al. 2007], así como la detección de la persona dominante [Hung & Gatica-Perez 2008], [Hung et al. 2008] a partir de la integración de la información de audio y vídeo adquirida por los sensores ubicados en una sala de reuniones. En la Figura 2.4 se muestra un esquema de la sala de reuniones implementada en el instituto IDIAP.

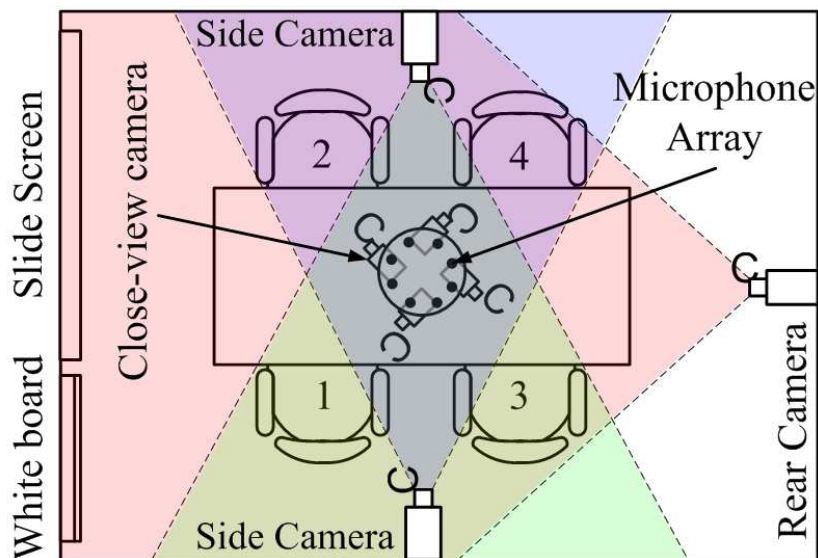


Figura 2.4. Esquema de la sala de reuniones implementada en el instituto IDIAP. Imagen extraída de [Hung et al. 2008].

En ninguno de los proyectos mencionados anteriormente se incorporan robots móviles controlados por el entorno. El uso de agentes controlados por el espacio para la interacción con los usuarios es estudiado por un número reducido de grupos de investigación.

Dentro del proyecto “sensor networked mobile robotics” del Departamento de Electrónica y Ciencias de la Computación de la Universidad Clemson se propone la obtención de un mapa de ocupación a partir de la fusión de la información adquirida por múltiples cámaras ubicadas en posiciones fijas [Hoover & Olsen 1999a] para, posteriormente planificar la trayectoria a seguir por un robot móvil [Hoover & Olsen 1999b], [Hoover & Olsen 2000]. En la Figura 2.5 se muestra el esquema de la red de sensores propuesta en el trabajo de Hoover y Olsen, así como el mapa de ocupación sobre el plano del suelo, a partir del cuál se planifica la trayectoria del robot móvil.

## Sensor Networked Robot

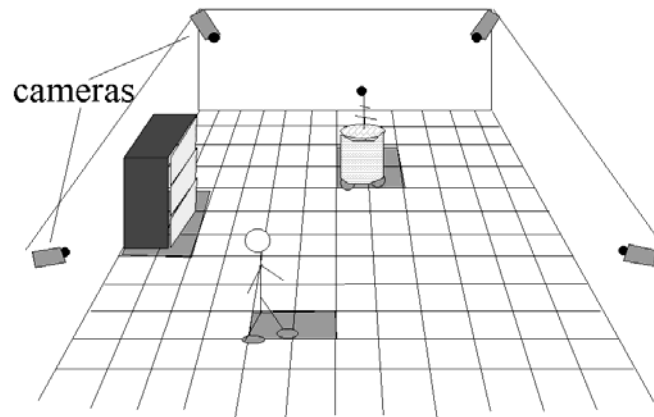


Figura 2.5. Esquema de la red de sensores propuesta por Hoover y Olsen [Hoover & Olsen 1999a], y mapa de ocupación sobre el plano del suelo.

Otra propuesta en esta línea es el proyecto MEPHISTO (A Modular and Extensible Path Planning System Using Observation) [Steinhaus et al. 1999], [Steinhaus et al. 2004] del Instituto de Diseño de Computadores y Tolerancia a Fallos de la Universidad de Karlsruhe, en Alemania. En este proyecto la inteligencia del sistema recae en los denominados LAPUs (*Local Area Processing Unit*) cada uno de los LAPUs se conecta a una Unidad de Control de Robots (RCU) la cuál se comunica con un robot mediante una conexión inalámbrica para enviarle los comandos, y recibir los mensajes de estado. La topología del sistema MEPHISTO se muestra en la Figura 2.6. En este trabajo, la detección de robots y obstáculos móviles se realiza mediante diferencia de imágenes mientras que, la localización se obtiene sin utilizar marcas artificiales, usando una descripción poligonal de cada uno de los robots.

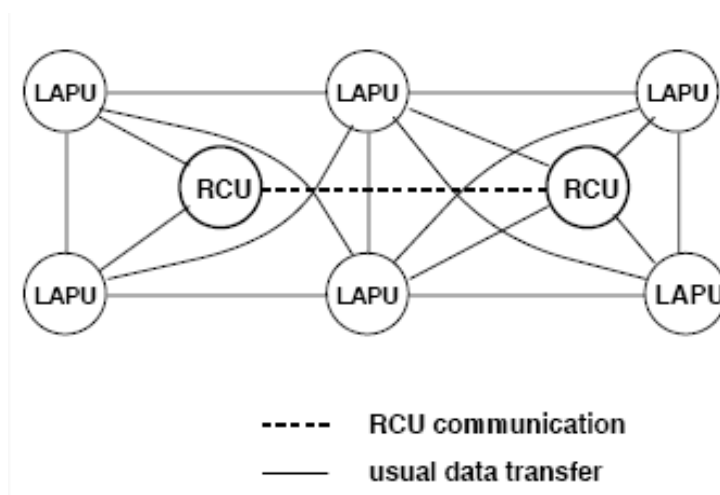


Figura 2.6. Topología de la arquitectura formada por LAPUs y RCUs propuesta en el sistema MEPHISTO

En la universidad de Tokio, Lee y Hashimoto [Lee & Hashimoto 2002] proponen un Espacio Inteligente en el que se desplazan robots y usuarios. El conjunto formado por un dispositivo de sensado (cámara o micrófono) junto con el sistema de procesamiento correspondiente se denomina DIND (*Distributed Intelligent Network Device*). La localización de los robots se realiza mediante marcas artificiales, mediante códigos de colores ubicados en barras, sobre los robots móviles. Una vez detectados los códigos de colores, es sencillo determinar la posición y orientación de los robots en el espacio inteligente. En el espacio inteligente desarrollado se realizan diferentes experimentos de navegación de robots móviles, así como de interacción entre robots móviles y usuarios [Niitsuma et al. 2009].

En los últimos años, dentro del Grupo de Espacios Inteligentes y Transporte (GEINTRA) de la Universidad de Alcalá se han desarrollado múltiples trabajos [Fernandez et al. 2007], [Pizarro et al. 2009] cuyo objetivo es la obtención de la posición de múltiples robots móviles en un espacio inteligente a partir de la información adquirida por un conjunto de sensores ubicados en posiciones fijas del entorno. Entre los sensores se incluyen diferentes alternativas tales como ultrasonidos, infrarrojos, voz y visión artificial.

La presente tesis se ha desarrollado dentro de las líneas de trabajo del grupo de investigación GEINTRA siendo el objetivo perseguido la obtención de la posición 3D de múltiples robots móviles, en un espacio inteligente, a partir de las imágenes adquiridas por una o varias cámaras sincronizadas y calibradas, que se encuentran ubicadas en posiciones fijas del espacio. Además, las cámaras están separadas entre sí, de forma que la distancia entre cámaras es comparable con la distancia entre cada cámara y los robots móviles. Esta configuración, conocida como *wide-baseline*, condiciona de forma notable las soluciones propuestas en este trabajo, ya que permite cubrir un área de interés mayor con un número de cámaras reducido, sin embargo, incrementa la dificultad de proceso de correspondencia de puntos entre las imágenes adquiridas por cada una de las cámaras.

Para la consecución de los objetivos perseguidos en esta tesis, es necesario llevar a cabo un conjunto de tareas:

- De forma previa a la localización es necesario detectar y segmentar tanto los robots móviles, como los posibles obstáculos en movimiento presentes en la

escena, obteniéndose como resultado los contornos de los objetos de interés en el plano imagen de cada una de las cámaras. Debido a las características del sistema propuesto: las cámaras se encuentran fijas, de forma que el fondo de la escena también permanece fijo, y se desea segmentar objetos en movimiento, la segmentación se obtiene en función de los parámetros de movimiento de los robots así como los posibles obstáculos móviles. A lo largo del apartado 2.2 se realiza una revisión de las técnicas de segmentación de movimiento propuestas en la literatura, prestando especial atención a las técnicas basadas en contornos activos por su interés para el desarrollo de la tesis.

- Tras la detección y segmentación de los objetos de interés, debe llevarse a cabo el posicionamiento de los mismos. Algunas de las alternativas propuestas en la literatura para la obtención de la posición 3D a partir de las imágenes adquiridas por cámaras externas se presentan a lo largo del apartado 2.3. Es importante mencionar que, con objeto de incrementar la precisión de los resultados, en esta etapa se incorpora toda la información disponible en el espacio inteligente, tanto la adquirida por las diferentes cámaras, como la proporcionada por el entorno (ya que, dado que los robots son agentes controlados por el espacio inteligente, se cuenta con información acerca de los comandos de guiado, así como de las medidas de los sensores odométricos de los mismos).

## **2.2. Segmentación de movimiento**

La segmentación de objetos en movimiento es una tarea fundamental para el análisis de secuencias de imágenes cuyo objetivo es agruparlas en conjuntos de píxeles cuyo movimiento en el plano imagen es coherente a lo largo de la secuencia. Todo ello sin tener un conocimiento previo acerca de qué píxeles de la imagen se mueven de acuerdo a un determinado modelo de movimiento.

El problema de la segmentación en función del movimiento ha sido estudiado en profundidad durante los últimos años, existiendo en la literatura numerosas técnicas que tratan de resolverlo. Dada la gran variedad de técnicas propuestas, no existe uniformidad a la hora de realizar una clasificación. En este trabajo, las diferentes

técnicas descritas se han organizado de acuerdo al principio en el que están basadas. Cabe destacar que a la hora de revisar las diferentes técnicas de segmentación de movimiento se ha prestado especial atención a las basadas en diferencia de imágenes, por su simplicidad y las basadas en flujo óptico, debido al interés que tienen para el desarrollo de esta tesis, mientras que el resto de técnicas se presentan de forma menos detallada.

### **2.2.1. Técnicas de segmentación de movimiento basadas en diferencia de imágenes**

La diferencia de imágenes es una de las técnicas más simples para la detección de cambios en las mismas. Estos métodos se usan principalmente para discriminar entre los grupos de píxeles que pertenecen al fondo estático, y aquellos que corresponden a objetos en movimiento. La principal ventaja de este tipo de técnicas es su bajo coste computacional, sin embargo no permiten obtener información compleja de la imagen.

La base de este tipo de métodos es el umbralizado de la intensidad de la imagen diferencia, obtenida al restar dos imágenes consecutivas píxel a píxel. Las técnicas de detección de movimiento basadas en diferencia de imágenes pueden dividirse en dos grupos, dependiendo de la imagen que se resta. Dichas imágenes pueden ser dos imágenes consecutivas de la secuencia, o bien la imagen a segmentar y una imagen de referencia en la que únicamente se tiene el fondo de la escena. En cualquiera de los casos, los resultados proporcionados por este tipo de técnicas se deterioran rápidamente en presencia de ruido en las imágenes, sombras o cambios de iluminación. Por este motivo no son adecuadas si se requiere que el sistema de segmentación sea robusto.

A modo de ejemplo, en la Figura 2.7(a) se presenta una secuencia de 4 imágenes de un robot. El resultado de umbralizar la diferencia de cada par de imágenes consecutivas se muestra en la Figura 2.7(b), mientras que en la Figura 2.7(c) se representa el resultado de umbralizar la diferencia de cada una de las imágenes de la Figura 2.7(a) y una imagen de referencia que únicamente contiene el fondo de la escena.

Esta técnica de segmentación de movimiento se utilizó en los años 70 para la codificación y compresión de secuencias de video [Haskell et al. 1972], aunque también



ha sido aplicada en otras áreas como la detección de movimiento en imágenes tomadas por satélite [Eghbali 1979], el seguimiento de personas [Segen 1996], etc.

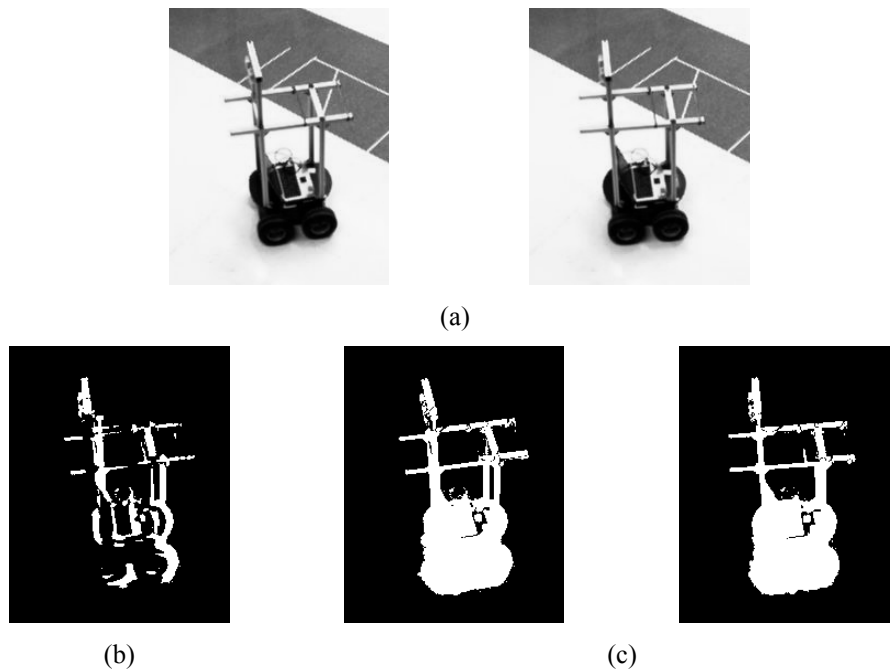


Figura 2.7. (a) Imágenes del robot en movimiento (b) Resultado de la umbralización de la diferencia del par de imágenes, calculada píxel a píxel (c) Resultado de la umbralización de la diferencia entre cada imagen de entrada y una imagen de referencia en la que únicamente aparece el fondo de la escena.

Existen diferentes trabajos en los que la detección de movimiento se realiza mediante sustracción del fondo. Con objeto de disminuir el efecto del ruido e incrementar la robustez, han sido propuestas diferentes alternativas. En [Gil & Pun, 1993] se propone un método multi-resolución que permite obtener una máscara alrededor de la zona ocupada por cada objeto móvil. Por otro lado, en [Renjie et al., 2007] se emplea un método de localización de objetos móviles en secuencias de imágenes mediante la diferencia de imágenes consecutivas. La disminución de la influencia del ruido se consigue dividiendo cada imagen de entrada en bloques uniformes y no superpuestos. Por cada uno de los bloques, se obtiene un valor en función de la intensidad media de los píxeles del bloque, y el número de píxeles del mismo cuya intensidad se encuentra por encima y por debajo del valor medio calculado. La imagen diferencia se calcula en función de la medida obtenida para cada uno de esos bloques.

Los métodos basados en la sustracción de una imagen del fondo estático de la escena, permiten simplificar el problema de segmentación. Sin embargo, presentan otras dificultades añadidas, como la necesidad de mantener actualizado el fondo de la escena frente a cambios de las características del propio fondo, o de las condiciones de

iluminación. En esta línea, en [Donohoe et al. 1988] se propone el uso de un filtro lineal para la actualización de cada uno de los píxeles de la imagen de fondo, mientras que en [Koller et al. 1994] se usa un filtro de Kalman con el mismo propósito. Otras alternativas que permiten incrementar la robustez frente al ruido de este tipo de técnicas de detección de movimiento se proponen en [Cheng and Chen 2006], donde se emplea la transformada Wavelet discreta para disminuir el efecto del ruido, o en [Colombari et al. 2007] cuyos autores proponen un modelo estadístico robusto del fondo de la escena.

### 2.2.2. Técnicas de segmentación de movimiento basadas en flujo óptico

El flujo óptico se define como el campo vectorial bidimensional de las velocidades aparentes en una imagen, que se obtiene a partir de una secuencia de imágenes, mediante el análisis de los cambios que sufre la luminancia en los puntos de las mismas. Hay que tener en cuenta que el flujo óptico no siempre se corresponde con el campo de movimiento en la imagen. Un ejemplo significativo es el puesto en [Horn & Schunck 1981] de una esfera rotando, que no tiene marcas en su superficie (textura) y bajo una iluminación constante que no produce cambios en la intensidad temporal de la imagen, a pesar de que exista movimiento en el espacio tridimensional. Por otro lado, también puede darse el caso contrario, en el que a pesar de que la esfera no se mueve, los cambios en la iluminación provocan cambios en la intensidad temporal de la imagen, de forma que, existe flujo óptico sin que haya movimiento 3D. Ambas situaciones se muestran en la imagen de la Figura 2.8.

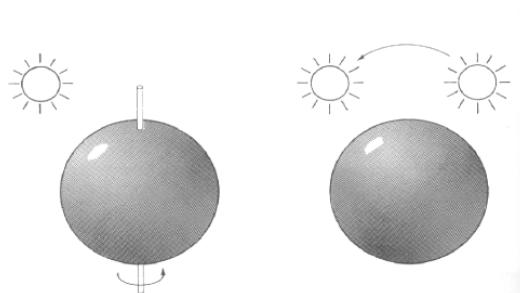


Figura 2.8. El flujo óptico no siempre se corresponde con el campo de movimiento. A la izquierda se muestra una esfera sin textura, bajo iluminación constante que rota sobre sí misma, sin que exista movimiento en la imagen. A la derecha se muestra una esfera fija, sometida a cambios en la iluminación, que provocan que exista un campo de movimiento (flujo óptico) en el plano imagen.

El flujo óptico ha sido ampliamente utilizado en aplicaciones de visión artificial, siendo inicialmente formalizado en [Horn & Schunck 1981]. Las técnicas de

segmentación de movimiento basadas en flujo óptico realizan la división de la imagen en diferentes regiones atendiendo a las discontinuidades en el flujo óptico, que se calcula a partir de la restricción de brillo [Horn & Schunck 1981]. Asumiendo que la intensidad de la imagen de cada punto de la misma no cambia respecto al tiempo, se cumple la expresión mostrada en la ecuación (2.1) en la que  $(\partial x_1, \partial x_2)$  representa el desplazamiento de la región  $(x_1, x_2, t)$  después de un tiempo  $\partial t$

$$I(x_1, x_2, t) \approx I(x_1 + \partial x_1, x_2 + \partial x_2, t + \partial t) \quad (2.1)$$

El desarrollo de Taylor alrededor de un punto  $(x, y)$  se obtiene mediante la siguiente expresión.

$$I(x + \partial x, y + \partial y, t + \partial t) = I(\mathbf{x}, t) + I_x(\mathbf{x}, t)dx + I_y(\mathbf{x}, t)dy + I_t(\mathbf{x}, t)dt + O(\delta^2) \quad (2.2)$$

Si las variaciones  $dx$ ,  $dy$  y  $dt$  son muy pequeñas, los términos  $O(\delta^2)$  pueden despreciarse, obteniendo la restricción de brillo definida en la ecuación (2.3).

$$I_{x_1}u + I_{x_2}v + I_t = 0 \quad (2.3)$$

En la ecuación (2.3)  $(I_{x_1}, I_{x_2})$  se corresponden con las componentes  $x$  e  $y$  del gradiente espacial de la intensidad de la imagen,  $I_t$  es la derivada temporal de la misma y  $(u, v)$  son las dos componentes del flujo óptico. Debido a que la restricción de brillo es una única ecuación con dos incógnitas  $(u, v)$ , son necesarias otras restricciones para poder calcular de forma unívoca el valor del flujo óptico en cada píxel de las imágenes de entrada. Esto es lo que se conoce como problema de apertura, y consiste en que, en una zona de la imagen sólo es posible medir la componente tangencial a dicho gradiente, a no ser que en dicha zona, o apertura, se encuentre una esquina, u otra componente con suficiente estructura en la intensidad que permita el cálculo de las dos componentes del flujo óptico, a partir de la restricción del mismo. Esto implica que, en general, no sea posible determinar la componente del flujo óptico perpendicular a la dirección del gradiente. En la Figura 2.9 se muestra de forma gráfica el problema de apertura.

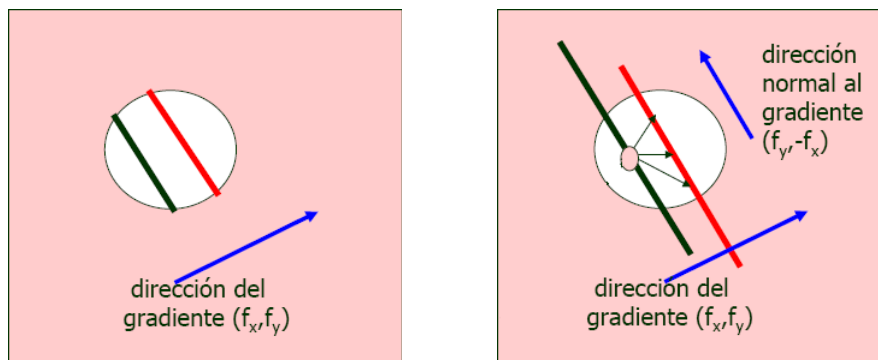


Figura 2.9. Representación gráfica del problema de apertura en el cálculo del flujo óptico.

En [Horn & Schunck 1981] se propone el uso de una segunda restricción, conocida como restricción de uniformidad, que, junto con la restricción de brillo definida anteriormente permite el cálculo del flujo óptico. Dicho cálculo se lleva a cabo mediante la minimización de la función de error mostrada en la ecuación (2.4), donde  $\alpha^r$  es una constante,  $\| \cdot \|$  representa la norma euclídea y  $\nabla$  es el operador gradiente.

$$\varepsilon^2 = \iint_{\text{Imagen}} (I_x u + I_y v + I_t)^2 dx dy + \alpha^r \iint_{\text{Imagen}} (\|\nabla u\|^2 + \|\nabla v\|^2) dx dy \quad (2.4)$$

Además del problema de apertura, existen otros problemas derivados del hecho de que las estimaciones del flujo óptico puedan ser ruidosas, especialmente en regiones con poca textura en la imagen. En [Black & Anandan 1991] se trata de disminuir el efecto del ruido usando algunas restricciones para interpolar el campo de flujo óptico. Por otro lado, dado que la restricción de brillo definida en la ecuación (2.3) ha sido obtenida considerando que las variaciones  $dx$ ,  $dy$  y  $dt$  son muy pequeñas, esta restricción es válida únicamente si el desplazamiento entre imágenes consecutivas de la secuencia es pequeño. En caso contrario, los términos de mayor orden deben ser tenidos en cuenta. En esta línea, en [Nagel 1983], el autor considera dichos términos para obtener una estimación más precisa del flujo óptico. Sin embargo, ninguna de las técnicas mencionadas proporciona la suficiente robustez frente a oclusiones, y regiones de poca textura en la imagen.

Uno de los problemas más significativos del método de [Horn & Schunck 1981] es que el término de regularización considerado suaviza por igual en todas las direcciones (es isotrópico) sin tener en cuenta las discontinuidades del flujo óptico.

Son muchos los trabajos realizados con el objetivo de modificar el término de regularización de la ecuación (2.4), de forma que se conserven las discontinuidades del flujo óptico tras el suavizado, destacando la propuesta de [Aubert et al. 1999], basada en

la sustitución de los términos  $\|\nabla w\|^2$  (donde  $w$  es  $u$  ó  $v$ ) por  $\phi(\|\nabla w\|)$  siendo  $\phi(\cdot)$  una función matemática denominada función de preservación de bordes, la cuál debe cumplir una serie de condiciones con objeto de preservar las discontinuidades del flujo óptico. En [Aubert et al. 1999], la estimación del flujo óptico se realiza mediante la resolución de un problema variacional, de forma que el campo de movimiento  $\mathbf{u} = (u, v)$  sobre el plano imagen será aquel que minimice la función definida en la ecuación (2.5) en la que  $\alpha^r$ , y  $\alpha^h$  son constantes reales y positivas,  $\Omega$  representa el dominio de la imagen, y  $c(x)$  es una función de penalización de las regiones homogéneas.

$$E(\mathbf{u}) = \int_{\Omega} (I_x u + I_y v + I_t)^2 dx dy + \alpha^r \int_{\Omega} (\phi(\|\nabla u\|) + \phi(\|\nabla v\|)) dx dy + \alpha^h \int_{\Omega} c(x) \|\mathbf{u}\|^2 dx dy \quad (2.5)$$

En [Aubert et al. 1999], la minimización de la ecuación (2.5) se realiza mediante el esquema semi-cuadrático (*half-quadratic*) [Geman & Reynolds 1992], demostrándose que el algoritmo de minimización converge.

Otra alternativa, también variacional, es la propuesta en [Mitiche & Sekkati 2006] y [Sekkati & Mitiche 2006a], donde se define una función objetivo para el caso de dos regiones (ecuación (2.6)) que está compuesta por cuatro términos.

$$E(\{\mathbf{e}_k\}_{k=1}^2, \gamma, u, v) = \sum_{k=1}^2 \int_{R_k} \{(\mathbf{d} \cdot \mathbf{e}_k)^2 + \mu (I_x u + I_y v + I_t)^2 + \nu (\phi(\|\nabla u\|) + \phi(\|\nabla v\|))\} dx + \lambda \int_{\gamma} ds \quad (2.6)$$

El primero de los términos es una restricción de movimiento sobre el plano imagen, dada por la expresión mostrada en la ecuación (2.7), en la que  $\mathbf{d}$  es un vector de dimensión 9 cuyos elementos dependen de las coordenadas sobre el plano imagen  $(x, y)$ , de las componentes del flujo óptico  $(u, v)$  y de la distancia focal de la cámara considerada, mientras que  $\mathbf{e}$  es el vector de parámetros esenciales, que están relacionados con los parámetros de movimiento rígido 3D  $(\mathbf{v}_c = (v_c^x \ v_c^y \ v_c^z)^T)$  y  $(\boldsymbol{\omega}_c = (\omega_c^x \ \omega_c^y \ \omega_c^z)^T)$ .

$$\mathbf{d} \cdot \mathbf{e} = 0 \quad (2.7)$$

Respecto al segundo término de la ecuación (2.6), éste coincide con la restricción de brillo de [Horn & Schunk 1981], mientras que los otros dos elementos de la ecuación son términos de regularización del flujo óptico, y las curvas que definen los contornos de la segmentación sobre el plano imagen. La minimización de la función objetivo se

lleva a cabo mediante un algoritmo iterativo formado por tres etapas que se repiten hasta alcanzar la convergencia.

En la misma línea, en [Sekkti & Mitiche 2006b] se define una función objetivo (ecuación (2.8)) cuya minimización permite estimar tanto los contornos de la segmentación de movimiento, como las componentes de velocidad 3D de los robots móviles, y el valor de la profundidad  $Z$ , definida como la distancia entre cada punto perteneciente a los robots y el plano imagen de la cámara. En este caso, la función propuesta está formada por tres términos. El primero de ellos relaciona la variaciones espacio-temporales de la intensidad de la imagen con los parámetros de movimiento 3D estimados en cada instante mediante la restricción de brillo 3D (definida [Sekkti & Mitiche 2006b]) mientras que las otras dos integrales son términos de regularización de la profundidad (mediante una función de preservación de bordes) y de las curvas que definen los contornos de la segmentación en el plano imagen.

$$E[\{\gamma_k\}_{k=1}^{N-1}, \{\mathbf{v}_{ck}\}_{k=1}^N, \{\boldsymbol{\omega}_{ck}\}_{k=1}^N, Z] = \sum_{k=1}^N \left[ \int_{\Omega_k} \psi_k^2(\mathbf{x}) d\mathbf{x} + \mu \int_{\Omega_k} g(\|\nabla Z\|) d\mathbf{x} \right] + \sum_{k=1}^{N-1} \lambda \oint_{\gamma_k} ds \quad (2.8)$$

$\lambda, \mu \in \mathfrak{R}$  y  $\lambda, \mu > 0$

Al igual que en el caso anterior, la minimización se lleva a cabo mediante un algoritmo *greedy* compuesto por tres etapas que se repiten hasta alcanzar la convergencia. Se considera que el algoritmo converge cuando los valores de las variables implicadas dejan de variar de forma significativa entre una iteración y la anterior.

Esta función objetivo es el punto de partida del trabajo desarrollado en esta tesis doctoral. Por ello, y dada la importancia que tiene, a lo largo del capítulo 4 se realiza un estudio exhaustivo de los diferentes elementos que la componen, así como del algoritmo de minimización y el criterio de convergencia considerado en esta tesis.

## 2.2.3. Otras técnicas de segmentación de movimiento

### 2.2.3.1. Basadas en capas (*layers*)

La segmentación del movimiento ha estado tradicionalmente ligada con la detección de movimiento, en la que cada región corresponde a un modelo particular de movimiento que refleja los cambios temporales en dicha región de la imagen. En esta

línea, en [Darrell & Pentland 1991] y [Wang & Adelson 1993] se propone una nueva representación conocida como representación en capas (*layered representation*) basada en múltiples modelos de movimiento soportadas en diferentes capas. Para cada una de las capas se calcula un modelo de movimiento, a partir de estos modelos de movimiento se actualizan las regiones de apoyo. La estimación de los modelos de movimiento, así como de sus regiones de apoyo se realiza habitualmente empleando el proceso iterativo EM (*Expectation – Maximization*) que alterna entre la segmentación de las medidas de la imagen (etapa E) y la estimación de los parámetros de movimiento (etapa M). En [Weiss 1996] se propone incorporar restricciones espaciales en al etapa E. Las propuestas basadas en EM tienen la ventaja de que proporcionan estimaciones robustas del movimiento combinando información de zonas amplias de la imagen, sin embargo, la convergencia a una solución óptima depende fuertemente de la inicialización.

En [Ke & Kanade 2004] se propone una solución robusta a los *outliers* para la obtención de un conjunto de *layers* que representan los diferentes modelos de movimiento sobre el plano imagen. Dicha solución explota el uso de una métrica basada en la distancia a los  $k$  vecinos más cercanos (*k-th nearest neighbor distance*, kNND), que permite eliminar los *outliers* para, posteriormente, obtener los diferentes *layers* mediante descomposición en valores singulares (SVD) de la matriz formada por los datos de entrada resultantes tras la eliminación de los *outliers*.

### **2.2.3.2. Basadas en factorización**

Otras propuestas para segmentación de movimiento son las basadas en factorización. Estas técnicas fueron inicialmente aplicadas en [Boult & Brown 1991] donde se propone una restricción de rango para estimar el número de movimientos independientes. La segmentación de los puntos de la imagen se obtiene a partir de los vectores singulares principales de la matriz de los puntos característicos en múltiples imágenes. Una técnica similar fue posteriormente empleada en [Costeira & Kanade 1998] donde, dado un conjunto de puntos, se define un matriz de interacción  $\mathbf{Q}$  (*shape interaction matrix*) que permite la agrupación de los puntos sin necesidad de estimar previamente el movimiento de los mismos. La principal desventaja de esta técnica es el hecho de que su funcionamiento se degrada rápidamente en presencia de ruido, debido a que la matriz  $\mathbf{Q}$  pierde rápidamente su capacidad discriminativa. En [Kanatani 2001] se propone una mejora mediante corrección de la dimensión y ajuste robusto, sin embargo

esta técnica no garantiza la obtención de la segmentación óptima global, debido a que está basada en una interacción punto-a-punto local. Por otra parte, en [Shi & Malik 2000] se demuestra que bajo ciertas suposiciones, el número de valores singulares diferentes de cero pueden usarse para determinar el número de movimientos en una imagen, mientras que, los movimientos pueden separarse empleando los vectores singulares adecuados, asociados a valores singulares no nulos.

En [Vidal & Sastry 2002], los autores proponen una solución algebraica para la segmentación de escenas estáticas o dinámicas a partir de la intensidad de las imágenes adquiridas mediante una cámara afín. Esta solución está basada en la restricción afín multicuerpo, que relaciona el movimiento de múltiples objetos con los cambios de intensidad en las imágenes generados por dicho movimiento. En este trabajo, tanto el número de objetos en movimiento, como los parámetros de dicho movimiento se obtienen mediante factorización. Dicha factorización se lleva a cabo mediante el análisis de componentes principales generalizado (GPCA) [Vidal et al. 2003], [Vidal et al. 2005]. En caso de tener múltiples vistas, GPCA también permite realizar la segmentación de movimiento, así como recuperar la estructura a partir del movimiento, mediante la factorización de la matriz epipolar multicuerpo, en caso de utilizar dos cámaras [Vidal et al. 2006] o del tensor trifocal multicuerpo [Hartley & Vidal 2004] para el caso de tres cámaras [Vidal & Hartley 2008]. Sin embargo, dada la naturaleza algebraica de GPCA, los resultados obtenidos no son satisfactorios en presencia de ruido en las derivadas parciales espaciales y temporal de las imágenes de entrada.

### **2.3. Posicionamiento de robots móviles a partir de las imágenes adquiridas por cámaras externas**

El objetivo del posicionamiento de robots móviles es la obtención de la posición y orientación de dichos robots, respecto al entorno en que se desplazan, con suficiente precisión. Como ya se ha comentado en la introducción a este capítulo, las diferentes alternativas para localización de robots móviles que existen en la literatura pueden dividirse en dos grandes grupos, en función de si los sensores se ubican a bordo de los robots, o en el entorno de movimiento de los mismos.



El sistema desarrollado en esta tesis pertenece al segundo grupo ya que la posición de los robots se obtiene a partir de las imágenes adquiridas por una, o varias cámaras calibradas, ubicadas en posiciones fijas del espacio inteligente. Cabe destacar además, que la configuración de las cámaras (*wide-baseline*), supone una importante restricción, ya que la distancia entre cámaras es grande, siendo comparable a la distancia entre cada cámara y los objetos de interés (robots móviles), lo que dificulta la obtención de la correspondencia entre las imágenes de las diferentes cámaras. Los diferentes métodos propuestos para obtener la posición 3D de robots (u otros objetos móviles) en las condiciones planteadas en esta tesis pueden dividirse en dos grandes grupos, en función del conocimiento a priori que es necesario de los robots. El primero de los grupos incluye aquellas técnicas que requieren un gran conocimiento acerca de los robots, lo cual se logra mediante la incorporación de marcas artificiales a los mismos, mientras que en el segundo se incluyen las técnicas en las que no se cuenta con información a priori de los robots, sino que se emplean sus características (marcas naturales), junto con la geometría de las cámaras para obtener la posición de los robots. A continuación, se describen brevemente algunos de los trabajos más significativos incluidos en cada uno de los dos grupos.

### **2.3.1. Posicionamiento utilizando marcas artificiales**

El uso de marcas artificiales está ampliamente extendido en aplicaciones de visión artificial, especialmente en trabajos relacionados con realidad aumentada y calibración, existiendo múltiples propuestas.

A la hora de diseñar las marcas artificiales, es importante que éstas cumplan un conjunto de condiciones que permitan que las marcas puedan ser determinadas de forma unívoca a partir de su proyección en el plano imagen de las cámaras (independientemente de la posición relativa entre las cámaras y las marcas), además, el proceso de detección de las mismas debe ser rápido y robusto.

El proceso de diseño y detección de marcas artificiales [Hoff et al. 1996], [Koller et al. 1997], [Kato et al. 2000] ha sido ampliamente estudiado para su uso en entornos de realidad aumentada, donde proporcionan resultados robustos, con suficiente velocidad y precisión.

Respecto a la navegación de robots móviles son diversos los trabajos que han utilizado marcas artificiales para el guiado de los mismos. Para llevar esto a cabo se pueden utilizar marcas artificiales ubicadas en puntos fijos del entorno y el sistema de captación a bordo de los robots (ello permite obtener la posición absoluta del robot respecto al entorno), o bien marcas artificiales a bordo de los robots y el sistema de captación en puntos fijos del entorno.

Dentro de la primera alternativa se han desarrollado diferentes trabajos, que presentan diversas alternativas para el diseño y la detección de las marcas artificiales. Así en [Briggs et al. 2000] se propone realizar la localización a partir de marcas formadas por un conjunto de barras en las que se incluye información codificada asociada a cada marca. Por otro lado, en el trabajo presentado en [Yang & Xinhe 2006] se emplean marcas artificiales en color, formadas por un conjunto de rectángulos simétricos y dos números definidos por 7 segmentos, como se muestra en el ejemplo presentado en la Figura 2.10(a).

También dentro del uso de marcas artificiales ubicadas en puntos fijos del entorno, en el trabajo de [Sooyong & Jae-Bok 2007] las marcas se realizan de un material que refleja la radiación infrarroja, de forma que sea sencillo detectarlas cuando la escena se ilumina con luz infrarroja. Las marcas empleadas tienen forma de triángulo equilátero como el mostrado en la Figura 2.10(b). La detección de los tres triángulos azules permite determinar la orientación, mientras que el resto de sectores se emplean para la identificación de las marcas.

En [Yu & Wenfei 2007], las marcas artificiales están formadas por una etiqueta RFID (*Radio Frequency Identification*) activada mediante un haz láser, a la que se incorpora un LED. De esta forma, el sistema de posicionamiento combina un lector de etiquetas RFID con un sistema de visión estéreo que permite la obtención de la posición 3D mediante triangulación.

Otro ejemplo de marcas artificiales, ubicadas en puntos fijos del entorno, empleadas para navegación de robots móviles son los códigos MR (*Mobile Robot codes*) [Feng et al 2009] formados por un pentágono equilátero dentro del cuál se incluye un código BCH (Figura 2.10(c)). Estas marcas presentan características invariantes frente a cambios en la iluminación.

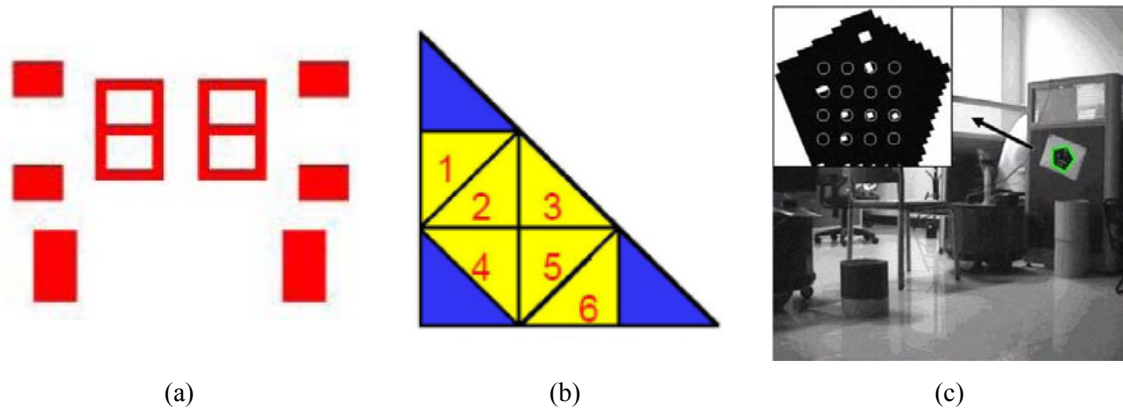


Figura 2.10. Ejemplo de las marcas artificiales utilizadas para la navegación de robots móviles (a) Marca formada por rectángulos simétricos y dos números de 7 segmentos [Yang & Xinhe 2006] (b) marca triangular empleada el trabajo de [Sooyong & Jae-Bok 2007] (c) Código MR (*Mobile Robot*) definido en [Feng et al 2009].

Cabe destacar que, las marcas artificiales ubicadas en el entorno también pueden emplearse para la localización de robots bajo el agua, tal como se muestra en [Zhang et al 2004]. Un ejemplo del tipo de marcas utilizadas en este trabajo se muestra en la Figura 2.11.

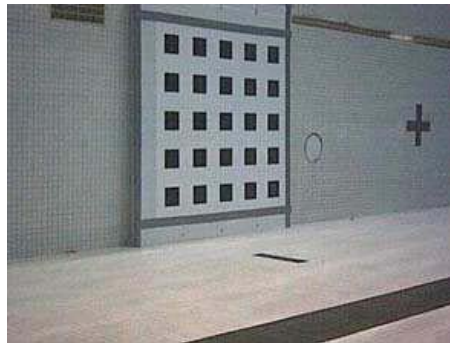


Figura 2.11. Marca artificial empleada en [Zhang et al. 2004] para la localización de un robot bajo el agua

Aunque en menor medida, las marcas artificiales también se emplean para el posicionamiento de robots en aplicaciones similares a la planteada en esta tesis. Esto es, las cámaras se ubican en el entorno de movimiento de los robots y las marcas a bordo de los robots. En este caso, los tipos de balizas empleados son también diversas.

Dentro de esta segunda alternativa, en [Lee et al. 2001] se propone el uso de barras que incluyen un código de color en las esquinas de los robots, de forma que la detección e identificación de estas barras permite determinar tanto la posición como la orientación de los robots. En [Huabin et al. 2004] las marcas empleadas son círculos en color, mientras que en [Fan et al. 2008], la identificación de los robots se realiza tras la detección de una marca formada por un conjunto de líneas y un triángulo, tal como se muestra en la Figura 2.12(a). La marca empleada en [Liang et al. 2008] para la identificación de los robots está compuesta por dos elementos de diferentes colores: un

rectángulo que identifica el robot y un cuadrado que define la orientación del mismo (Figura 2.12(b)).

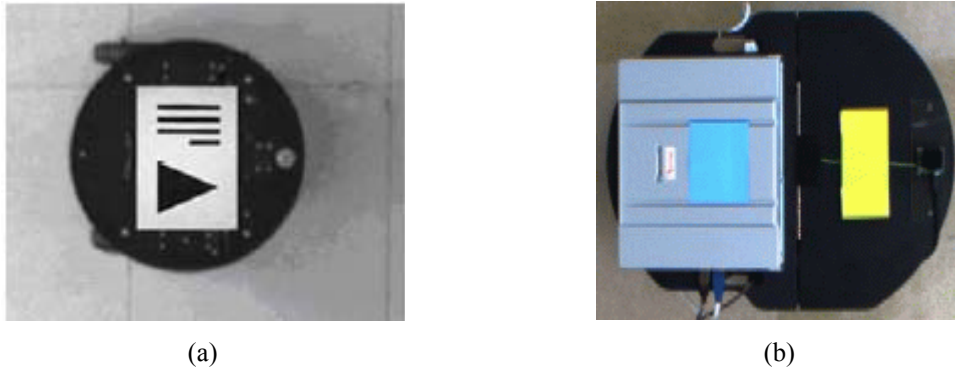


Figura 2.12. Marcas artificiales empleadas sobre un robot móvil para su localización e identificación (a) marca utilizada en [Fan et al. 2008]. (b) marca utilizada en [Liang et al. 2008].

Por otro lado, en [Fernandez et al. 2007], el balizamiento del robot se realiza empleando una estructura en forma de T, que consiste en cuatro elementos circulares de 2 centímetros de diámetro formados por diodos de infrarrojos (IR) (Figura 2.13). Esta marca puede ser detectada con facilidad incluso en situaciones adversas de iluminación. Además, la distribución de los elementos permite recuperar de forma sencilla tanto la posición como la orientación del robot, una vez que la marca ha sido detectada.



Figura 2.13. Array de diodos de infrarrojo empleados para el posicionamiento y guiado de un robot móvil en [Fernandez et al. 2007].

Finalmente, mencionar que el uso de marcas artificiales puede emplearse también para la localización de otros objetos móviles, incluso si éstos no se desplazan por el suelo. Este es el caso del trabajo de [Coelho & Campos 1999] donde se plantea el uso de una marca artificial formada por tres círculos que permite determinar la posición y orientación de un dirigible.

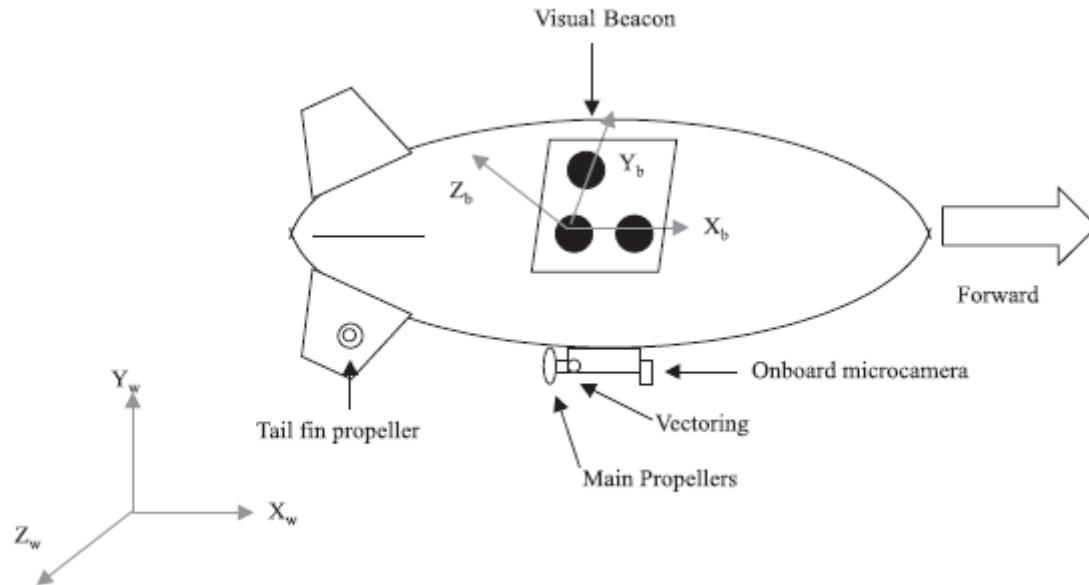


Figura 2.14. Esquema de funcionamiento del sistema de guiado de un dirigible mediante marcas artificiales descrito en [Coelho & Campos 1999].

El uso de marcas artificiales para la detección y localización de robots móviles permite obtener resultados adecuados tanto en aplicaciones en las que las cámaras se ubican a bordo de los robots, como en aquellas que emplean cámaras externas. Sin embargo, estas técnicas presentan diversos inconvenientes que no las hacen adecuadas para aplicaciones como la de interés de esta tesis.

En primer lugar, la necesidad de incorporar las marcas artificiales sobre el robot limita notablemente la capacidad de detección y localización del sistema, ya que únicamente es posible detectar los robots que incluyan las citadas marcas, eliminando la posibilidad de localizar otros robots u obstáculos móviles que puedan existir en el espacio inteligente. Además, las técnicas descritas anteriormente no hacen uso de la información contenida en las imágenes, ya que únicamente se centra en la detección de las marcas.

### 2.3.2. Posicionamiento utilizando marcas naturales

Con objeto de suplir las deficiencias que presentan las soluciones basadas en marcas artificiales, surgen los sistemas de visión artificial basados en marcas naturales, que hacen uso de la información debida a las características geométricas de los elementos de la escena, así como del movimiento de las imágenes, tratando de extraer la máxima información posible de las mismas.

Los primeros trabajos en el área de posicionamiento de objetos mediante marcas naturales requieren el uso de un modelo de los objetos a localizar. En esta alternativa el objetivo es la detección de un conjunto de características en la imagen (que pueden ser de naturaleza diversa: puntos, líneas, parámetros de movimiento, etc.) que puedan ajustarse al modelo definido previamente. Conocida la calibración de las cámaras y las características métricas del modelo, es posible recuperar la posición 3D de los objetos. Estos trabajos pueden dividirse en función de las características que definen el modelo empleado.

Las primeras aproximaciones en este área están basadas en la detección de bordes que permiten definir la imagen mediante un conjunto de primitivas geométricas (en general segmentos de recta y curvas) que posteriormente se ajustan al modelo dado. Algunos trabajos en esta línea pueden encontrarse en [Lowe 1992] o [Koller et al. 1993]. En la actualidad, este tipo de técnicas apenas se emplean debido a las restricciones geométricas que imponen al objeto a detectar.

Con objeto de reducir estas restricciones aparecen una serie de trabajos basados en el uso de puntos característicos sobre el plano imagen, en lugar de rectas o curvas [DeMenthon and Davis 1995], [Christy & Horaud 1999]. El uso de puntos presenta diversas ventajas, tales como el incremento de la robustez de los algoritmos frente a cambios de iluminación u oclusiones.

Otras alternativas son las basadas en modelos de movimiento. Estas técnicas están fuertemente relacionadas con los trabajos realizados en el área de segmentación de movimiento que han sido descritos en el apartado 2.2, así como con los algoritmos desarrollados a lo largo de esta tesis. En la mayoría de las técnicas propuestas, el flujo óptico se recupera mediante el método de Lucas y Kanade [Lucas & Kanade 1981].

Los diferentes métodos descritos hasta ahora requieren un alto grado de conocimiento previo de los robots, ya que se basan en el uso de modelos a priori de los mismos. En los últimos años, han surgido nuevas técnicas que permiten recuperar la posición de los robots sin la necesidad de ese conocimiento a priori.

Entre estas técnicas se encuentra el trabajo de [Hsieh et al. 2009] donde se propone para la obtención de la posición de los robots la integración, mediante un filtro de partículas, de las medidas de los sensores odométricos de los robots y la información de las imágenes adquiridas por un conjunto de cámaras externas calibradas.

También en esta línea se enmarca la propuesta de [Pizarro et al. 2009] que permite obtener la pose de un robot móvil a partir de las imágenes adquiridas por un conjunto de cámaras ubicadas en posiciones fijas del espacio inteligente. El sistema planteado se divide en dos fases: en la primera de ellas se obtiene un modelo tridimensional del robot, así como su pose inicial para, posteriormente, en la segunda fase, actualizar la pose cada vez que se adquiere una nueva imagen.

## 2.4. Conclusiones

En este capítulo se han expuesto algunos de los trabajos realizados por la comunidad científica en el área de interés de esta tesis, prestando especial atención a las diferentes técnicas que permiten la segmentación y localización en condiciones similares a las planteadas en esta tesis, es decir, a partir de imágenes adquiridas por cámaras fijas en el entorno.

Tal como se ha podido observar a lo largo de este capítulo, las primeras propuestas para la segmentación y el posicionamiento de robots móviles se basaban en el uso de sensores a bordo de los mismos. Sin embargo, en la actualidad se aprecia una tendencia al uso de redes de sensores externos, y al desarrollo de espacios inteligentes, debido a las ventajas que proporciona el uso de este tipo de soluciones. Por este motivo, este trabajo se enmarca en un área de creciente interés en el que trabajan numerosos grupos de investigación.

Son diversas las alternativas propuestas en la literatura para la resolución de los problemas planteados en esta tesis: la obtención de la segmentación de movimiento y la estimación de la posición 3D de objetos móviles a partir de la información adquirida por cámaras calibradas que se ubican en posiciones fijas del entorno de movimiento. Sin embargo, no existe una solución totalmente satisfactoria que dé respuesta a las demandas de robustez y flexibilidad que se requiere en este tipo de aplicaciones.

En la presente tesis doctoral, que se ha desarrollado dentro de las líneas de investigación del grupo GEINTRA de la Universidad de Alcalá, se propone una nueva alternativa para la segmentación de movimiento y la estimación de la posición 3D de un número variable de robots móviles en un espacio inteligente, sin necesidad de tener un conocimiento a priori de dichos robots.





## Capítulo 3

# **Espacio Inteligente (ISPACE-UAH)**



El trabajo desarrollado en esta tesis se enmarca dentro del área de los espacios inteligentes, entendiendo como espacio inteligente un área física dotada de sensores (cámaras, micrófonos, etc.) que permiten percibir lo que ocurre en el entorno. Estos sensores son controlados por un sistema de supervisión dotado de capacidad de análisis y toma de decisiones. Dicho sistema de supervisión también controla un conjunto de actuadores que le permiten interactuar con el entorno y los usuarios del mismo. Entre esos actuadores se encuentran los robots móviles (agentes controlados) cuya segmentación y posicionamiento son el objeto de esta tesis.

El uso de sensores ubicados en el entorno de movimiento de los robots en lugar de a bordo de los mismos presenta importantes ventajas. En primer lugar, al trasladar la inteligencia al entorno, la electrónica a bordo de los robots puede reducirse, con las ventajas que ello conlleva. Además, la navegación simultánea de múltiples robots no requiere incrementar la complejidad de la infraestructura. Por otro lado, la información adquirida por los sensores externos es más completa que la que puede obtener cada uno de los robots de forma individual; esto permite obtener información acerca de todos los robots, facilitando la cooperación entre ellos.

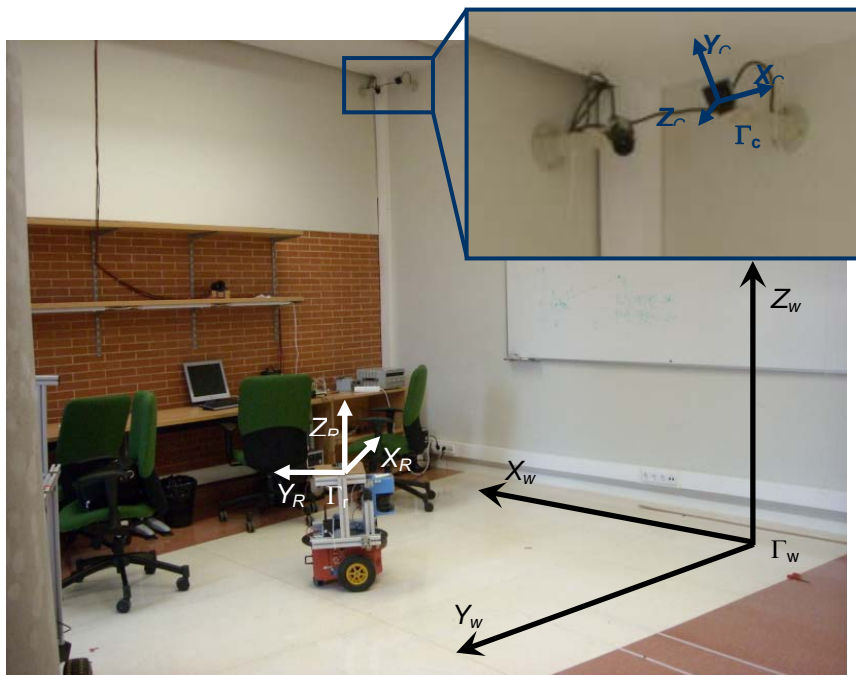


Figura 3.1. Espacio inteligente del Departamento de Electrónica de la UAH (ISPACE-UAH).

En este capítulo se presenta el espacio inteligente instalado en Departamento de Electrónica de la Universidad de Alcalá (ISPACE-UAH) en el que se ha desarrollado esta tesis (al que pertenece la foto mostrada en la Figura 3.1). Para comenzar, en el apartado 3.1 se describe la estructura del espacio inteligente, así como los diferentes

elementos incluidos en el mismo. Además, se describe brevemente la arquitectura hardware y software del sistema de visión del ISPACE-UAH. Debido a que en este trabajo las medidas se representan en diferentes sistemas de coordenadas, se ha considerado necesario describir en el apartado 3.2 esos sistemas de coordenadas, así como las relaciones entre ellos.

### 3.1. Estructura del espacio inteligente

En el caso concreto de esta tesis, el sistema sensorial está formado por un conjunto de  $n_c$  cámaras calibradas. Estas cámaras están ubicadas en posiciones fijas del entorno y distribuidas estratégicamente de forma que sea posible cubrir toda el área de movimiento de los robots, y con un número reducido de cámaras. Todas las cámaras se encuentran conectadas a una red de comunicaciones que permite que la adquisición de las mismas se realice de forma sincronizada, siendo controlada por el sistema de supervisión del espacio inteligente. Este sistema de supervisión controla además los actuadores que le permiten interactuar con los usuarios. En el caso bajo estudio los actuadores son un conjunto de robots que se conectan al sistema de supervisión a través de una red inalámbrica. En la Figura 3.2 se muestran, de forma esquemática los diferentes elementos del espacio inteligente considerado en esta tesis.

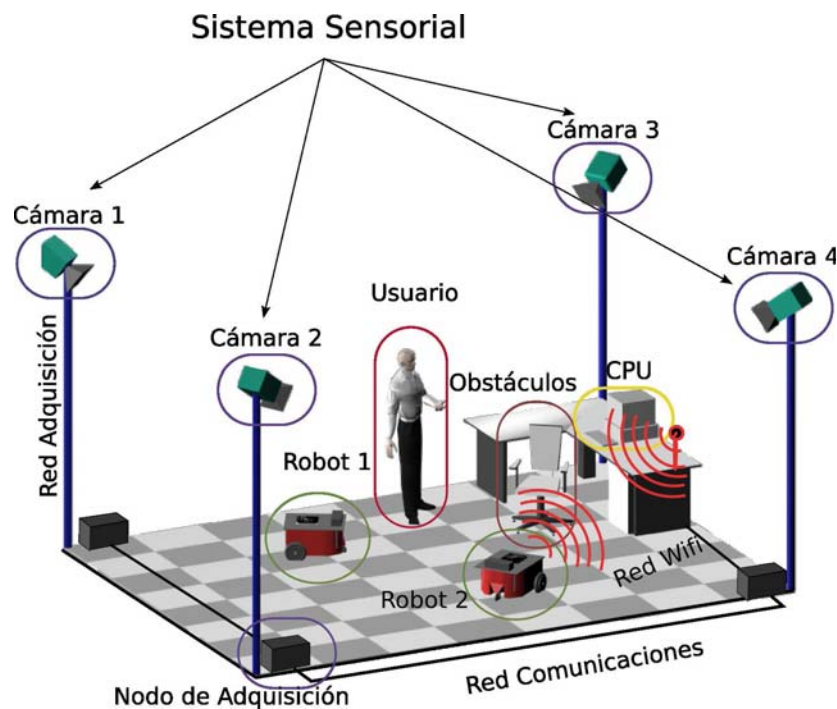


Figura 3.2. Estructura general del ISPACE-UAH considerado en esta tesis en la que se muestran los diferentes elementos existentes en el mismo.

### 3.1.1. Arquitectura hardware del ISPACE-UAH

A continuación se describen brevemente los elementos que forman la arquitectura hardware del sistema de cámaras del ISPACE-UAH:

- Cámaras con sensor CCD con una resolución de 640x480 píxeles. La distancia focal es de 6.5 milímetros, con un campo de visión (FOV) de aproximadamente 45°. Cada una ellas se conecta a un nodo de procesamiento a través de un bus *Firewire* (IEEE1394) que permite una adquisición de imágenes en formato RGB a una velocidad de 15 fps. Entre ellas están sincronizadas mediante un trigger externo.
- Nodos de adquisición y procesamiento: son plataformas PC de propósito general que incluyen puertos *Firewire* para la conexión con las cámaras y así como *Gigabit Ethernet* para la conexión a la red local (LAN). En esta tesis se considera que cada uno de los nodos se conecta a una única cámara, a pesar de tener la capacidad de controlar y procesar la información de múltiples cámaras.
- Infraestructura de red de área local (LAN) que incluye una conexión inalámbrica a través de la que se conectan los robots móviles y una conexión cableada que une los diferentes nodos de procesamiento.
- Un nodo cliente encargado de procesar, en conjunto, la información recibida de los nodos de procesamiento y los robots móviles. El nodo cliente, en que se realizarán las diferentes pruebas experimentales presentadas a lo largo de esta tesis, es un PC con un procesador Intel Core 2 CPU 6600 a 2.40GHz, con 3,50 GB de memoria RAM. Cabe mencionar aquí que todos los experimentos han sido llevados a cabo utilizando Matlab.
- Robots móviles enlazados vía radio con el nodo cliente capaces de recibir comandos de guiado y enviar información de sus sensores odométricos. Todos los robots utilizados en los experimentos presentados en esta tesis son P3-DX de Pioneer. Estos robots, de dimensiones 44.5x40x24.5 cm, están provistos de un sistema de tracción diferencial formado por dos ruedas, e incluyen controladores de bajo nivel para cada una de las ruedas, así como sensores odométricos y una plataforma PC embebida con hardware de conexión inalámbrica IEEE 802.11.

### 3.1.2. Arquitectura software de ISPACE-UAH

La arquitectura software está basada en un sistema cliente-servidor que usa conexiones TCP/IP. Tanto los nodos de procesamiento como los PCs embebidos en los robots móviles actúan como servidores. Cada uno de los nodos se encarga de pre-procesar las imágenes y enviar los resultados a un cliente. Los PCs de los robots permiten recibir los comandos de control de un cliente, así como enviar las lecturas de sus sensores odométricos. Por otro lado, el cliente es el encargado de procesar, en conjunto, la información recibida de los nodos de procesamiento y los robots móviles. En la Figura 3.3 se muestra un diagrama general de la arquitectura hardware y software del ISPACE-UAH.

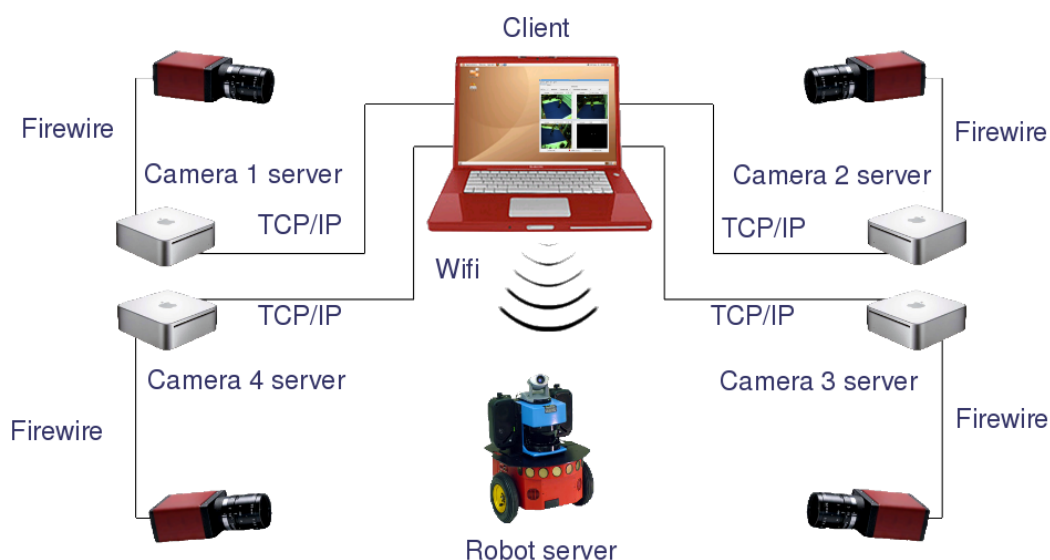


Figura 3.3. Diagrama general de la arquitectura hardware y software del ISPACE-UAH

## 3.2. Sistemas de referencia de coordenadas en el ISPACE-UAH

A lo largo de esta tesis, las coordenadas de los puntos 3D,  $\mathbf{P} = (X, Y, Z)^T$  se expresan respecto a diferentes sistemas de coordenadas 3D en el ISPACE-UAH. Por este motivo se ha considerado importante describir aquí cada uno de estos sistemas de referencia, así como las relaciones entre ellos.

En la Figura 3.4 se muestran de forma esquemática los sistemas de referencia que se utilizan en esta tesis.

El más importante es el sistema de coordenadas global (representado en color rojo en la figura) con respecto al que se representan todas las medidas relativas al entorno de movimiento de los robots. Este sistema de coordenadas global se denomina sistema de referencia del mundo ( $\Gamma_w$  siguiendo la terminología anglosajona) y su origen de coordenadas se encuentra sobre el plano del suelo. Cabe destacar que a lo largo de esta tesis, este sistema de referencia se denominará indistintamente sistema de referencia global o sistema de referencia del mundo, aunque en todos los casos se representará usando el símbolo  $\Gamma_w$ .

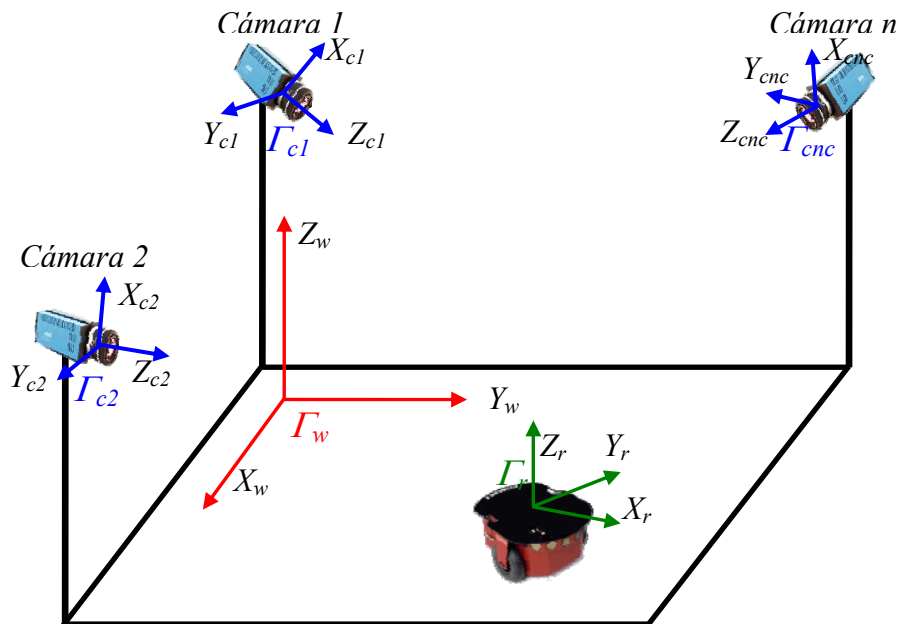


Figura 3.4. Sistemas de referencia en el ISPACE-UAH. Los ejes de coordenadas del sistema de referencia del mundo ( $\Gamma_w$ ) se representan en color rojo, mientras que los ejes del sistema de referencia de cada cámara ( $\Gamma_{ci}$ ,  $i = 1, 2, \dots, n_c$ ) se representan en color azul, y los ejes del sistema de referencia del robot en color verde.

Los sensores odométricos de cada uno de los  $N-1$  robots móviles proporcionan medidas acerca de la posición relativa de los mismos. Por este motivo es necesario definir un sistema de referencia por cada robot, ubicado sobre el citado robot y solidario con el mismo, al que denominamos  $\Gamma_{ri}$  ( $i=1, 2, \dots, N-1$ ) y que se representa en color verde en la Figura 3.4.

Además, por cada una de las cámaras se define un sistema de coordenadas de referencia, representado en color azul en la Figura 3.4, cuyo origen coincide con el centro óptico de cada cámara (ubicado en el centro geométrico del plano focal) y cuyo eje  $Z_{ci}$  es el eje óptico de la misma, tal como se puede apreciar en el esquema mostrado en la Figura 3.5.

A lo largo de esta tesis, el sistema de referencia de cada cámara se denominará  $\Gamma_{ci}$  ( $i=1,2,\dots,n_c$ ) donde el subíndice  $i$  identifica a la cámara respecto a la que se toma el sistema de coordenadas. En algunos apartados en los que sólo se considera una cámara (o bien el proceso desarrollado es el mismo para cada una de las cámaras) se prescinde del subíndice  $i$ , denominando al sistema de referencia de la cámara únicamente  $\Gamma_c$ .

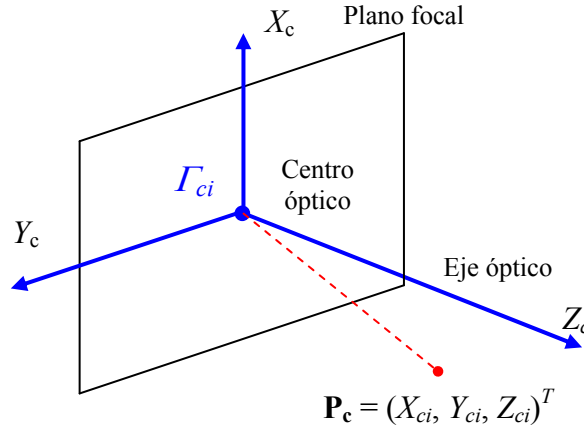


Figura 3.5. Sistema de referencia 3D de la cámara  $i$  ( $\Gamma_{ci}$   $i=1,2,\dots,n_c$ ) en la que el origen de coordenadas se ubica en el centro óptico y el eje  $Z_{ci}$  coincide con el eje óptico de la cámara.

El proceso de calibración de las cámaras permite obtener las relaciones entre el sistema de referencia de cada una de ellas y el plano imagen de las mismas (a través de los parámetros intrínsecos), así como entre cada cámara y el sistema de referencia global  $\Gamma_w$ , mediante los parámetros extrínsecos.

### 3.2.1. Geometría de formación de la imagen: modelo de cámara pin-hole

Se considera que las cámaras utilizadas pueden modelarse como una lente *pin-hole*. Éste es el modelo ideal más sencillo y describe la cámara como una caja completamente cerrada, en uno de cuyos lados se practica una apertura que es atravesada por los rayos emitidos o reflejados por el objeto, formando una imagen 2D invertida y de menor tamaño a una distancia  $f$  según se muestra en el esquema de la Figura 3.6(a), donde  $f$  es la distancia focal de la lente.

Además, la proyección del objeto invertido sobre un plano a una distancia  $f$  del centro de la cámara es, matemáticamente equivalente, a la proyección de dicho objeto sobre un plano situado a una distancia  $-f$ . Esto se muestra de forma gráfica en Figura 3.6(b).



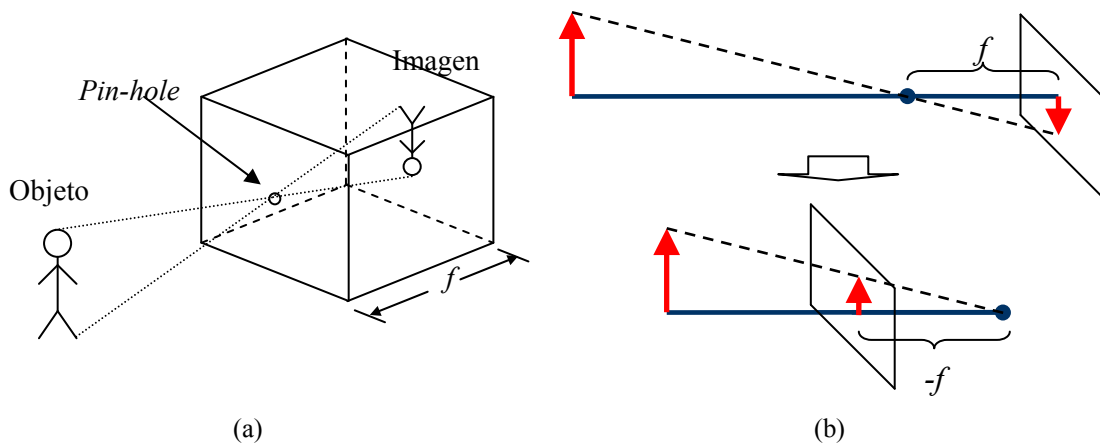


Figura 3.6. (a) Modelo de formación de la imagen en una cámara *pin-hole* representada como una caja cerrada con un agujero en uno de sus lados. (b) Proyección equivalente: la proyección invertida de un objeto en el plano imagen a una distancia  $f$  del centro óptico de la cámara es equivalente a la proyección de dicho objeto en un plano situado a una distancia  $-f$  del centro de la cámara.

A continuación se define el modelo geométrico de una cámara *pin-hole*, así como las ecuaciones que definen dicho modelo, y los parámetros extrínsecos e intrínsecos de la misma. En todas las menciones realizadas al sistema de referencia de la cámara a lo largo de este apartado se prescinde del subíndice  $i$  debido a que el modelo descrito puede aplicarse a cualquiera de las  $n_c$  cámaras. Para comenzar, en la Figura 3.7 se muestra el modelo geométrico de una cámara *pin-hole*.

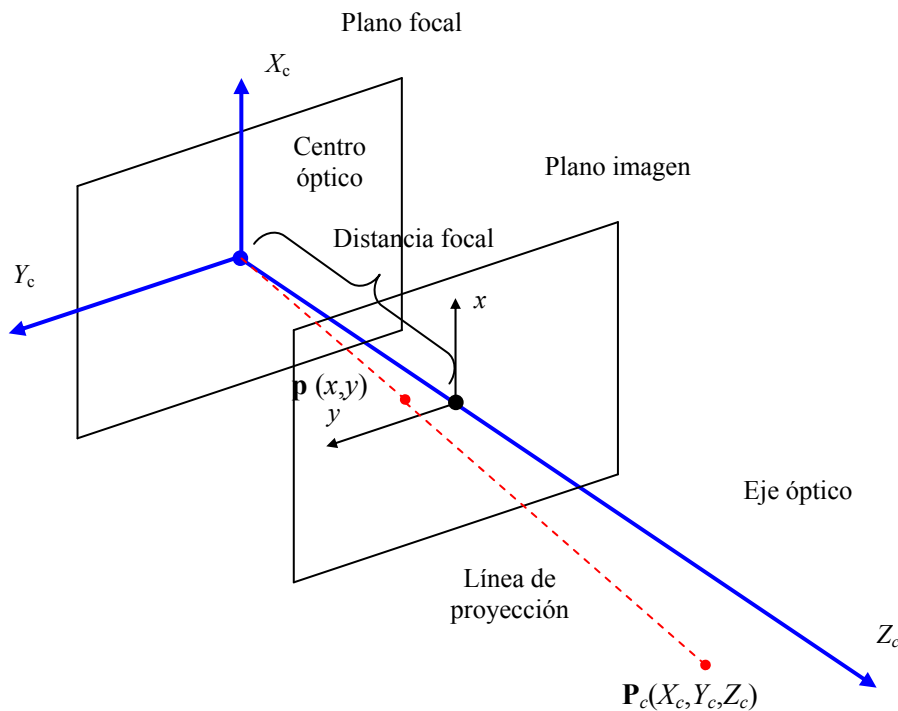


Figura 3.7. Modelo geométrico de cámara *pin-hole*. En color azul se representan los ejes del sistema de referencia de la cámara ( $\mathcal{I}_c$ ) mientras que en color negro se muestran los ejes de coordenadas sobre el plano imagen  $(x, y)$  y en línea roja discontinua la línea de proyección del punto 3D  $\mathbf{P}_c(X_c, Y_c, Z_c)$  sobre el plano imagen en el píxel  $\mathbf{p}(x, y)$ .

El modelo de cámara pin-hole permite obtener la proyección de los puntos del espacio 3D sobre el plano imagen 2D de forma que un punto 3D  $\mathbf{P}_c(X_c, Y_c, Z_c)$ , representado en el sistema de referencia de la cámara  $\Gamma_c$ , se proyecta sobre el plano imagen en el punto 2D  $\mathbf{p}(x, y)$  pasando a través del centro óptico ( $C$ ), localizado en el centro del plano focal. La recta que une al punto  $\mathbf{P}_c$  y el centro óptico de la cámara se denomina línea de proyección, e intersecta el plano imagen en el píxel  $\mathbf{p}(x, y)$ , que es la proyección del punto 3D  $\mathbf{P}_c(X_c, Y_c, Z_c)$ .

Considerando este modelo de proyección, las coordenadas del píxel sobre el plano imagen  $\mathbf{p}(x, y)$  pueden expresarse en función de las coordenadas del punto 3D en  $\Gamma_c$  ( $\mathbf{P}_c(X_c, Y_c, Z_c)$ ) en función de la distancia focal de la cámara ( $f$ ) utilizando las ecuaciones (3.1) y (3.2).

$$x = f \frac{X_c}{Z_c} \quad (3.1)$$

$$y = f \frac{Y_c}{Z_c} \quad (3.2)$$

En una cámara real, el sistema de coordenadas global  $\Gamma_w$ , está relacionado con el de la cámara  $\Gamma_c$  y el plano imagen a través de un conjunto de parámetros físicos como son la distancia focal, el tamaño de píxel, la posición y orientación de las cámaras, etc. Estos parámetros pueden dividirse en dos tipos: intrínsecos y extrínsecos.

### 3.2.1.1. Parámetros intrínsecos

Los parámetros intrínsecos son aquellos que dependen únicamente de la cámara, y permiten relacionar las coordenadas 3D en el sistema de referencia de la cámara, con las coordenadas de los píxeles 2D sobre el plano imagen. Estos parámetros incluyen la distancia focal, la localización del origen de coordenadas en el plano imagen, el tamaño efectivo del píxel en unidades métricas y los coeficientes de distorsión de la lente.

Con objeto de expresar las coordenadas sobre el plano imagen en unidades de píxeles, es necesario introducir un factor de escala en cada una de las dimensiones, en función del tamaño efectivo de los píxeles, en las ecuaciones (3.1) y (3.2). Si denominamos  $K_x$  y  $K_y$  a los factores de escala en  $x$  e  $y$  respectivamente, se obtienen las ecuaciones (3.3) y (3.4). En general, los valores de  $K_x$  y  $K_y$  son diferentes debido a que los píxeles son rectangulares, y no cuadrados.

$$x = K_x f \frac{X_c}{Z_c} \quad (3.3)$$

$$y = K_y f \frac{Y_c}{Z_c} \quad (3.4)$$

Por otro lado, en general, el origen del sistema de coordenadas sobre el plano imagen no se suele ubicar en el centro de dicho plano, sino que se sitúa en una de las esquinas del mismo. Esto obliga a tener que añadir a las ecuaciones anteriores un desplazamiento ( $s_1$  y  $s_2$ ) que define la posición del centro del sistema de coordenadas en el plano imagen.

$$x = K_x f \frac{X_c}{Z_c} + s_1 \quad (3.5)$$

$$y = K_y f \frac{Y_c}{Z_c} + s_2 \quad (3.6)$$

Finalmente, es posible que, debido al proceso de fabricación, los ejes del sistema de coordenadas sobre el plano imagen no formen un ángulo  $\theta$  de  $90^\circ$ . En este caso, las ecuaciones (3.5) y (3.6) se transforman en:

$$x = K_x f \frac{X_c}{Z_c} - K_x f \cot \theta \frac{Y_c}{Z_c} + s_1 \quad (3.7)$$

$$y = \frac{K_y f}{\sin \theta} \frac{Y_c}{Z_c} + s_2 \quad (3.8)$$

Estas ecuaciones pueden escribirse de forma compacta utilizando la expresión (3.9) en la cual  $\mathbf{K}_c$  es la matriz de parámetros intrínsecos. Cabe resaltar que en este caso se ha utilizado una representación homogénea tanto para las coordenadas 2D sobre el plano imagen como las coordenadas 3D del punto en  $\Gamma_c$ .

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \frac{1}{Z_c} \mathbf{K}_c \cdot \mathbf{P}_c = \frac{1}{Z_c} \begin{pmatrix} K_x f & K_x f \cot \theta & s_1 & 0 \\ 0 & \frac{K_y f}{\sin \theta} & s_2 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \quad (3.9)$$

En caso de que el ángulo entre los ejes de coordenadas sea de  $90^\circ$ , el término  $K_x f \cot \theta$  es nulo, mientras que  $\sin \theta = 1$ , de forma que los términos introducidos en la

ecuación anterior no afectan a la matriz de transformación, resultando la matriz ideal mostrada en la ecuación (3.10).

$$\mathbf{K}_c = \begin{pmatrix} K_x f & 0 & s_1 & 0 \\ 0 & K_y f & s_2 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} f_x & 0 & s_1 & 0 \\ 0 & f_y & s_2 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad \begin{matrix} f_x = K_x \cdot f \\ f_y = K_y \cdot f \end{matrix} \quad (3.10)$$

### 3.2.1.2. Parámetros extrínsecos

Los parámetros extrínsecos de la cámara describen la relación espacial entre el sistema de referencia de esa cámara ( $\Gamma_c$ ), y el sistema de referencia del mundo ( $\Gamma_w$ ) mediante una matriz de rotación y un vector de traslación. En la Figura 3.8 se muestran de forma esquemática el sistema de referencia del mundo ( $\Gamma_w$ ) y el de una de las cámaras ( $\Gamma_c$ ).

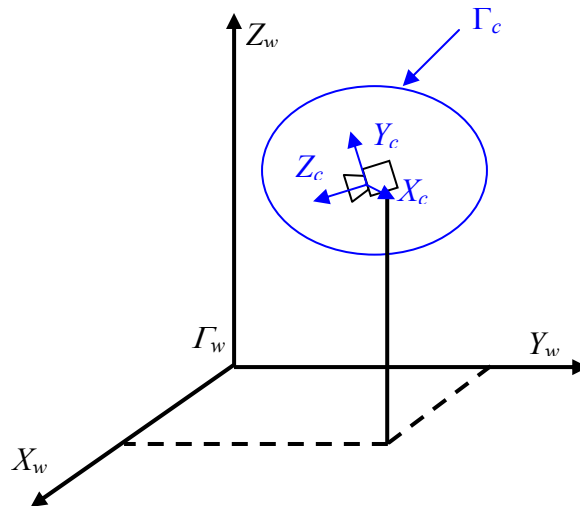


Figura 3.8. Relación entre sistemas de referencia. En color negro se representan los ejes del sistema de referencia global (del mundo)  $\Gamma_w$  mientras que en azul se muestran los ejes del sistema de referencia de una cámara ( $\Gamma_c$ ).

Dadas las coordenadas de un punto 3D  $\mathbf{P}_w = (X_w, Y_w, Z_w)^T$ , respecto al sistema de referencia global  $\Gamma_w$ , las coordenadas de dicho punto respecto al sistema de referencia de la cámara  $\mathbf{P}_c = (X_c, Y_c, Z_c)^T$  se obtienen mediante la expresión (3.11)

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R}_{wc(3 \times 3)} & \mathbf{t}_{wc(3 \times 1)} \\ \mathbf{0}_{(1 \times 3)} & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \quad (3.11)$$

donde  $\mathbf{R}_{wc}$  es la matriz de rotación (3x3) ortonormal que relaciona ambos sistemas de referencia y  $\mathbf{t}_{wc}$  es el vector de traslación (3x1) que proporciona información de las

coordenadas del centro de proyección de la cámara, respecto del sistema de referencia  $\Gamma_w$ .

Sustituyendo la expresión de  $\mathbf{P}_c$  de la ecuación (3.11) en la ecuación (3.9) se obtiene la siguiente relación entre las coordenadas en píxeles sobre el plano imagen y las coordenadas 3D en  $\Gamma_w$ .

$$\mathbf{P}_c = \frac{1}{Z_c} \mathbf{M} \mathbf{P}_w \quad \text{con} \quad \mathbf{M} = \mathbf{K}_c \cdot [\mathbf{R}_{wc} \quad \mathbf{t}_{wc}] \quad (3.12)$$

El proceso de calibración de las cámaras permite la obtención de los parámetros intrínsecos y extrínsecos de la misma. Cabe destacar que, a lo largo de esta tesis, estos parámetros se consideran conocidos a priori, ya que han sido obtenidos previamente, con suficiente precisión, mediante un proceso de calibración.



## Capítulo 4

# **Segmentación de movimiento y posicionamiento 3D mediante la minimización de una función objetivo**





De forma general, el objetivo de esta tesis es la obtención de la segmentación, en función de su movimiento, y la posición 3D de múltiples robots móviles, de forma precisa, a partir de las imágenes captadas por una o múltiples cámaras, ubicadas en posiciones fijas del espacio en que se desplazan los robots.

Siguiendo la idea presentada en el trabajo de [Sekkati & Mitiche 2006b], tanto la segmentación como la posición 3D se obtienen tras la minimización de una función objetivo. Esta función objetivo incluye información acerca de diferentes parámetros relacionados con el movimiento de los robots sobre el plano imagen, así como las velocidades y posiciones de los puntos pertenecientes a los robots móviles en el sistema de referencia de cada cámara ( $\Gamma_c$ ).

Para comenzar, en el apartado 4.1 se describe en detalle el caso más sencillo, en el que la segmentación de movimiento se realiza a partir de la información adquirida por una única cámara. En esta situación, la función objetivo, a partir de cuya minimización se obtiene la segmentación de movimiento, es la propuesta en el trabajo de [Sekkati & Mitiche 2006b], con las diferencias derivadas de las características propias de la aplicación estudiada en esta tesis, tales como el uso de cámaras fijas, y la configuración *wide-baseline* de dichas cámaras (la distancia entre cámaras y robots es grande).

Las diferentes variables implicadas en la función objetivo propuesta en [Sekkati & Mitiche 2006b], así como el proceso de minimización de dicha función se describen con detalle a lo largo del apartado 4.1. En este apartado también se exponen las debilidades de esta técnica, debidas fundamentalmente al uso de una única cámara para la segmentación y posicionamiento.

Dado que en el ISPACE-UAH se cuenta con la información adquirida por múltiples cámaras, calibradas y sincronizadas entre sí, se ha realizado una búsqueda de técnicas que permitan la incorporación de la información adquirida por esas cámaras al resultado final, con objeto de incrementar la robustez y precisión de los resultados del algoritmo de segmentación y posicionamiento.

En esta línea, la propuesta más inmediata es el uso de la técnica descrita en el apartado 4.1 por cada una de las cámaras disponibles y la posterior fusión de los resultados obtenidos por cada una de esas cámaras, tal como se describe en el apartado 4.1.4 de este capítulo. Esta técnica tiene algunas ventajas, como la posibilidad de realizar el procesado de forma independiente por cada cámara. Sin embargo, el proceso

de inicialización de la profundidad debe hacerse de forma conjunta para todas las cámaras. Además, el cálculo de la segmentación y posición 3D por cada cámara y la posterior fusión de resultados no permiten cumplir los requerimientos de robustez de esta tesis, ya que durante el proceso de minimización de la función objetivo de cada cámara, no se contempla la información adquirida por el resto de cámaras.

En el apartado 4.2 se presenta la segunda alternativa, en este caso, en lugar de realizar la fusión de datos tras la minimización de la función objetivo, se ha propuesto una nueva función objetivo en la que se incorpora la información adquirida por múltiples cámaras. De este modo, tras la minimización de la función objetivo para múltiples cámaras propuesta en esta tesis, se obtienen la segmentación de movimiento, los parámetros de movimiento 3D (componentes de velocidad lineal y angular) y la posición 3D en el sistema de referencia global del espacio inteligente ( $\Gamma_w$ ) de forma robusta. Esto es así porque se incorpora la información de todas las cámaras disponibles.

## **4.1. Segmentación de movimiento y posicionamiento 3D a partir de las imágenes adquiridas por una cámara**

En este apartado se presenta la función objetivo para una cámara. Para ello se comienza definiendo la restricción de brillo 3D (que se obtiene a partir de la restricción de brillo constante sobre el plano imagen) para posteriormente definir la función objetivo y exponer el proceso de minimización de dicha función.

### **4.1.1. Restricción de brillo 3D para objetos rígidos**

En este apartado se describe el proceso de obtención de la restricción de brillo 3D para objetos rígidos a partir de la restricción de brillo [Sekki & Mítiche 2006b] mostrada en la ecuación (4.1). Esta restricción se obtiene de forma independiente para cada una de las cámaras, estando las coordenadas de los puntos 3D representadas en el sistema de referencia de la cámara correspondiente ( $\Gamma_{ci}$ ). Dado que este proceso únicamente se describe para una de las cámaras, a lo largo de este apartado se prescinde del subíndice  $i$  que identifica a la cámara.

$$I_x \cdot u + I_y \cdot v + I_t = 0 \quad (4.1)$$

Sea  $\mathbf{P}$  un punto perteneciente a un sólido rígido en movimiento cuyas coordenadas 3D en el sistema de referencia de la cámara  $\Gamma_c$  son:

$$\mathbf{P}_c = (X_c \quad Y_c \quad Z_c)^T \quad (4.2)$$

La velocidad del punto  $\mathbf{P}$  en el sistema de referencia de la cámara puede expresarse utilizando la ecuación (4.3) en la que  $\mathbf{v}_c = (v_c^x \quad v_c^y \quad v_c^z)^T$  y son las componentes de velocidad lineal y angular, respectivamente, en  $\Gamma_c$ .

$$\dot{\mathbf{P}}_c = \begin{pmatrix} \dot{X}_c \\ \dot{Y}_c \\ \dot{Z}_c \end{pmatrix} = \mathbf{v}_c + \boldsymbol{\omega}_c \times \mathbf{P}_c \Rightarrow \begin{pmatrix} \dot{X}_c \\ \dot{Y}_c \\ \dot{Z}_c \end{pmatrix} = \begin{pmatrix} v_c^x + \omega_c^y Z_c - \omega_c^z Y_c \\ v_c^y + \omega_c^z X_c - \omega_c^x Z_c \\ v_c^z + \omega_c^x Y_c - \omega_c^y X_c \end{pmatrix} \quad (4.3)$$

Por otro lado, la ecuación de proyección de las cámaras que se muestra en la ecuación (4.4) permite obtener las coordenadas del punto  $\mathbf{P}$  sobre el plano imagen  $(x, y)$  a partir de las coordenadas de dicho punto en  $\Gamma_c$ . En (4.4)  $\lambda$  es la constante de proporcionalidad de las coordenadas homogéneas.

$$\begin{pmatrix} \lambda x \\ \lambda y \\ \lambda \end{pmatrix} = \begin{pmatrix} f_x & 0 & s_1 \\ 0 & f_y & s_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} \quad \begin{matrix} f_x = K_x f \\ f_y = K_y f \end{matrix} \quad (4.4)$$

Derivando la expresión (4.4) respecto del tiempo se obtienen las componentes de velocidad de  $\mathbf{P}$  sobre el plano imagen  $(\dot{x}, \dot{y})$  :

$$\begin{pmatrix} \dot{\lambda} x + \lambda \dot{x} \\ \dot{\lambda} y + \lambda \dot{y} \\ \dot{\lambda} \end{pmatrix} = \begin{pmatrix} f_x \dot{X}_c + s_1 \dot{Z}_c \\ f_y \dot{Y}_c + s_2 \dot{Z}_c \\ \dot{Z}_c \end{pmatrix} \quad (4.5)$$

Denominando  $u$  y  $v$  a las componentes de velocidad sobre el plano imagen en el eje  $x$  e  $y$  respectivamente, es decir:  $u = \dot{x}$  y  $v = \dot{y}$ , de la ecuación (4.5) se obtiene:

$$\begin{cases} u = \dot{x} = \frac{1}{Z_c} (f_x \dot{X}_c - (x - s_1) \dot{Z}_c) \\ v = \dot{y} = \frac{1}{Z_c} (f_y \dot{Y}_c - (y - s_2) \dot{Z}_c) \end{cases} \quad (4.6)$$

Si se sustituyen en la ecuación (4.6) las componentes de la velocidad en  $\Gamma_c$   $(\dot{X}_c \ \dot{Y}_c \ \dot{Z}_c)^T$  por las expresiones obtenidas en la ecuación (4.3), se obtiene:

$$\begin{cases} u = \frac{1}{Z_c} (f_x v_c^x - (x - s_1) v_c^z) - \frac{Y_c (x - s_1)}{Z_c} \omega_c^x + \left( f_x + \frac{X_c (x - s_1)}{Z_c} \right) \omega_c^y - f_x \frac{Y_c}{Z_c} \omega_c^z \\ v = \frac{1}{Z_c} (f_y v_c^y - (y - s_2) v_c^z) - \left( f_y + \frac{Y_c (y - s_2)}{Z_c} \right) \omega_c^x + \frac{X_c (y - s_2)}{Z_c} \omega_c^y + f_y \frac{X_c}{Z_c} \omega_c^z \end{cases} \quad (4.7)$$

De la ecuación de proyección (4.4) se tiene que  $x = s_1 + f_x \frac{X_c}{Z_c}$  e  $y = s_2 + f_y \frac{Y_c}{Z_c}$ , de forma que las coordenadas 3D  $(X_c \ Y_c)$  en  $\Gamma_c$  pueden expresarse en función de las coordenadas en el plano imagen  $(x, y)$ :

$$X_c = \frac{Z_c}{f_x} (x - s_1), \quad Y_c = \frac{Z_c}{f_y} (y - s_2) \quad (4.8)$$

Sustituyendo las expresiones de  $X_c$  e  $Y_c$  (de la ecuación (4.8)) en la ecuación (4.7) y reorganizando los términos se obtienen las expresiones dadas por (4.9) de las componentes de velocidad del punto  $\mathbf{P}$  sobre el plano imagen  $(u, v)$  en función de las coordenadas de dicho punto sobre la imagen  $(x, y)$ , de la coordenada  $Z_c$  y de las componentes de velocidad lineal y angular en  $\Gamma_c$   $(\mathbf{v}_c, \boldsymbol{\omega}_c)$ :

$$\begin{cases} u = \frac{1}{Z_c} (f_x v_c^x - (x - s_1) v_c^z) - \frac{(x - s_1)(y - s_2)}{f_y} \omega_c^x + \left( f_x + \frac{(x - s_1)^2}{f_x} \right) \omega_c^y - \frac{f_x}{f_y} (y - s_2) \omega_c^z \\ v = \frac{1}{Z_c} (f_y v_c^y - (y - s_2) v_c^z) - \left( f_y + \frac{(y - s_2)^2}{f_y} \right) \omega_c^x + \frac{(x - s_1)(y - s_2)}{f_x} \omega_c^y + \frac{f_y}{f_x} (x - s_1) \omega_c^z \end{cases} \quad (4.9)$$

Teniendo en cuenta las expresiones de  $u$  y  $v$  en la restricción de brillo constante dadas por (4.10)

$$I_x u + I_y v + I_t = 0 \quad (4.10)$$

donde  $[I_x, I_y, I_t]$  son las derivadas parciales espacial y temporal de la imagen, se obtiene la siguiente expresión para la restricción de brillo 3D para objetos rígidos:

$$\begin{aligned} I_t + \frac{1}{Z_c} (f_x I_x v_c^x + f_y I_y v_c^y - ((x - s_1) I_x + (y - s_2) I_y) v_c^z) \\ + \left( -f_y I_y - \frac{(y - s_2)}{f_y} ((x - s_1) I_x + (y - s_2) I_y) \right) \omega_c^x \\ + \left( f_x I_x + \frac{(x - s_1)}{f_x} ((x - s_1) I_x + (y - s_2) I_y) \right) \omega_c^y \\ + \left( -\frac{f_x}{f_y} (y - s_2) I_x + \frac{f_y}{f_x} (x - s_1) I_y \right) \omega_c^z = 0 \end{aligned} \quad (4.11)$$

La ecuación (4.11) se pueden escribir de forma más compacta utilizando notación matricial, tal como se muestra en (4.12)

$$I_t + \mathbf{s} \frac{\mathbf{v}_c}{Z_c} + \mathbf{q} \boldsymbol{\omega}_c = 0 \quad (4.12)$$

En la ecuación (4.12)  $\mathbf{v}_c = (v_c^x \ v_c^y \ v_c^z)^T$  y  $\boldsymbol{\omega}_c = (\omega_c^x \ \omega_c^y \ \omega_c^z)^T$  son las componentes de velocidad lineal y angular del punto  $\mathbf{P}$  en  $\Gamma_c$ , y  $\mathbf{s}$  y  $\mathbf{q}$  son los vectores fila de tres componentes mostrados en la ecuación (4.13), y que dependen únicamente de las derivadas parciales espaciales y temporal de las imágenes de entrada  $[I_x, I_y, I_t]$ , de las coordenadas en el plano imagen del punto  $\mathbf{P}$   $(x, y)$  y de las distancias focales de la cámara  $(f_x, f_y)$ .

$$\mathbf{s} = \begin{pmatrix} f_x I_x \\ f_y I_y \\ -(x-s_1)I_x - (y-s_2)I_y \end{pmatrix}^T, \quad \mathbf{q} = \begin{pmatrix} -f_y I_y - \frac{y-s_2}{f_y} ((x-s_1)I_x + (y-s_2)I_y) \\ -f_x I_x - \frac{x-s_1}{f_x} ((x-s_1)I_x + (y-s_2)I_y) \\ -\frac{f_x}{f_y} (y-s_2)I_x + \frac{f_y}{f_x} (x-s_1)I_y \end{pmatrix}^T \quad (4.13)$$

#### 4.1.2. Función objetivo para una cámara

En la ecuación (4.14) se presenta la función objetivo propuesta en [Sekhati & Mitiche 2006b]. La minimización de esta función permite obtener la segmentación de movimiento, así como la localización 3D (a falta de un factor de escala debido al uso de una única cámara) de los diferentes robots que se desplazan por el espacio inteligente.

Para el uso de la ecuación (4.14) en esta tesis, ha sido necesario realizar un proceso de ajuste de algunos de los parámetros, con el fin de conseguir las mayores prestaciones y teniendo presentes las condiciones cambiantes del espacio de movimiento, así como la distribución *wide-baseline* de las cámaras.

Cabe destacar aquí que, en las ecuaciones presentadas a lo largo de todo el apartado 4.1 se ha prescindido del subíndice “c” debido a que se considera una única cámara. Por este motivo se sobrentiende que todas las variables están referidas al sistema de referencia de dicha cámara.

Según se puede observar, la función objetivo (4.14) depende de tres grupos diferentes de parámetros:

- Las curvas  $\{\gamma_k\}_{k=1}^{N-1}$  que definen los contornos de los  $(N-1)$  robots móviles presentes en la escena dividiendo la imagen en  $N$  regiones.
- Las componentes de la velocidad lineal y angular de cada uno de los  $(N-1)$  robots móviles, así como del fondo de la escena, en el sistema de referencia de la cámara:  $\{\mathbf{v}_{ck}\}_{k=1}^N, \{\boldsymbol{\omega}_{ck}\}_{k=1}^N$ .
- La distancia de cada punto 3D ( $\mathbf{P}$ ) a la cámara, que coincide con la coordenada  $Z_c$  de dicho punto en el sistema de referencia de la cámara  $\Gamma_c$ .

$$E[\{\gamma_k\}_{k=1}^{N-1}, \{\mathbf{v}_{ck}\}_{k=1}^N, \{\boldsymbol{\omega}_{ck}\}_{k=1}^N, Z] = \sum_{k=1}^N \left[ \int_{\Omega_k} \psi_k^2(\mathbf{x}) d\mathbf{x} + \mu \int_{\Omega_k} g(\|\nabla Z\|) d\mathbf{x} \right] + \sum_{k=1}^{N-1} \lambda \oint_{\gamma_k} ds \quad (4.14)$$

$$\lambda, \mu \in \mathfrak{R} \text{ y } \lambda, \mu > 0$$

En la ecuación (4.14)  $\psi_k(\mathbf{x})$  es la restricción de brillo 3D para los píxeles de la región  $k$  (definida en la ecuación (4.12)),  $\lambda$  y  $\mu$  son constantes reales y positivas que ponderan la contribución de cada uno de los términos y  $\nabla = (\partial_x, \partial_y)$  es el operador gradiente espacial. Hay que tener en cuenta que tanto la restricción de brillo 3D ( $\psi_k(\mathbf{x})$ ) como la profundidad ( $Z$ ) dependen de las coordenadas de cada píxel en la imagen representadas mediante el vector  $\mathbf{x} = (x, y)$  en la ecuación. Mientras que la última integral está definida sobre cada una de las curvas que definen el contorno de los robots móviles, representada por las variable  $\mathbf{s}$ .

En esta función objetivo, la primera integral mide la conformidad de la interpretación 3D de las variaciones espacio-temporales de la secuencia en cada una de las  $N$  regiones a través de la restricción de brillo 3D (definida en la ecuación (4.12)). Las otras dos integrales de la ecuación son términos de regularización de la profundidad, así como de las  $N-1$  curvas que definen los contornos de las  $N$  regiones en que se divide la imagen ( $N-1$  correspondientes, en nuestro caso, a los robots móviles y una correspondiente al resto de la escena).

Cabe destacar que la función objetivo mostrada en la ecuación (4.14) coincide, en cuanto a su forma, con un contorno activo de Mumford-Shah [Mumford & Shah 1989].

Mediante la minimización de la función (4.14) es posible obtener tanto la segmentación de movimiento de los robots móviles presentes en la escena, como las componentes de velocidad lineal y angular y la posición 3D de dichos robots en  $\Gamma_c$ . Dado que la función objetivo depende de tres grupos de parámetros, en [Sekkati &

Mitiche 2006b] se propone realizar esta minimización dentro de un proceso *greedy* e iterativo formado por tres etapas que se repiten hasta alcanzar la convergencia. En cada una de estas tres etapas dos de los grupos de parámetros se suponen fijos y conocidos, y se obtiene el tercero. De forma previa al comienzo del proceso iterativo es necesario inicializar tanto las curvas  $N-1$  que definen los contornos de las  $N$  regiones  $\{\gamma_k\}_{k=1}^{N-1}$  como la distancia de cada píxel al sistema de referencia de la cámara ( $Z_c$ ).

A continuación, en el apartado 4.1.3 se describe el algoritmo de minimización de la función objetivo. Posteriormente se presenta el proceso que se ha propuesto en esta tesis para la inicialización de las curvas que definen los contornos de los robots móviles presentes en la escena. También se incluye un estudio para el ajuste de las constantes  $\lambda$  y  $\mu$  que ponderan la contribución de cada uno de los términos de la función objetivo, así como para la elección de la función de preservación de bordes.

### **4.1.3. Minimización de la función objetivo**

Para la minimización de la función objetivo se emplea un algoritmo *greedy* e iterativo de tres etapas, que se describen a lo largo del apartado 4.1.3.1. En cada una de las etapas del algoritmo de minimización se suponen fijos y conocidos dos de los tres grupos de parámetros de los que depende la función objetivo mostrada en la ecuación (4.14) y se obtiene el tercer grupo.

Tras la inicialización de la profundidad y las curvas que definen los contornos de los posibles robots móviles en el plano imagen, las tres etapas se repiten hasta alcanzar la convergencia. Se considera que el algoritmo converge cuando los valores de los grupos de variables implicados dejan de evolucionar de forma significativa. El criterio de convergencia adoptado en este trabajo se expone en el apartado 4.1.3.2.

Tanto el número de iteraciones hasta alcanzar la convergencia (y consecuentemente el tiempo de cómputo del algoritmo) como la precisión de los resultados obtenidos dependen fuertemente de los valores iniciales de las curvas y la profundidad, así como de los valores elegidos para las constantes  $\lambda$  y  $\mu$ , que ponderan la contribución de cada uno de los términos a la función objetivo. Debido a esta dependencia, se ha realizado un estudio exhaustivo que ha permitido realizar el ajuste de los valores de las constantes  $\lambda$  y  $\mu$  para diferentes funciones de preservación de bordes ( $g(a)$ ), que se presenta en el

capítulo 7. Por otro lado, a lo largo del capítulo 5 se analiza en profundidad el proceso de inicialización de las curvas que definen los contornos de las regiones en que se segmenta la imagen, mientras que en el capítulo 6 se describe la inicialización de la profundidad, además del algoritmo utilizado para la estimación del número de robots móviles presentes en cada imagen de la secuencia.

#### 4.1.3.1. Etapas del proceso de minimización

En la Figura 4.1 se representan, a modo de resumen, las diferentes etapas del algoritmo de minimización de la función objetivo para segmentación de movimiento rígido 3D. Cada una de las tres etapas que conforman el algoritmo se describe brevemente a continuación.

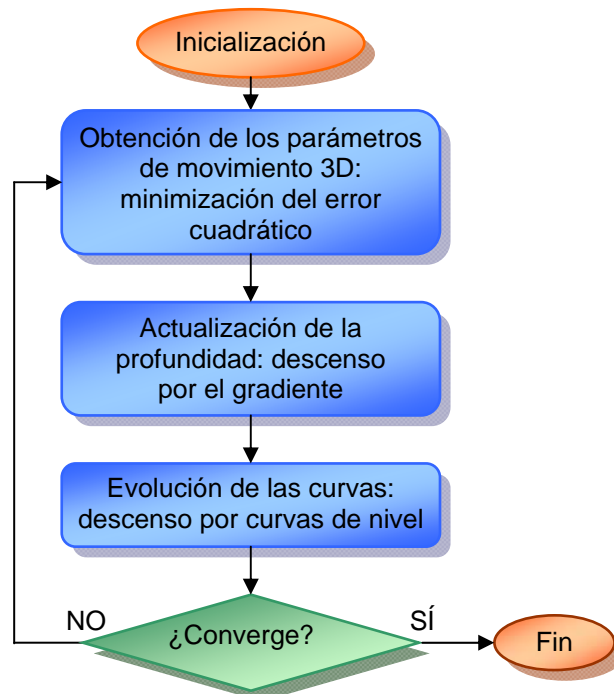


Figura 4.1. Diagrama de bloques general del proceso de minimización de la función objetivo definida en la ecuación (4.14).

#### Obtención de los parámetros del movimiento 3D

En esta primera etapa del algoritmo se suponen conocidas tanto las curvas que definen los contornos de los  $n$  robots móviles presentes en la escena, como el valor de la profundidad de cada uno de los puntos proyectados sobre el plano imagen (coordenada  $Z$  en  $\Gamma_c$ ). De esta forma, la función objetivo a minimizar se reduce a:

$$E\left(\{\mathbf{v}_{ck}\}_{k=1}^N, \{\boldsymbol{\omega}_{ck}\}_{k=1}^N\right) = \sum_{k=1}^N \int_{\Omega_k} \psi_k^2(\mathbf{x}) d\mathbf{x} \quad (4.15)$$



Según se indica en [Sekkati & Mitiche 2006b], debido a que la restricción de brillo para objetos rígidos 3D ( $\psi_k$ ) depende linealmente de las componentes de velocidad en el sistema de referencia de la cámara,  $\mathbf{v}_{ck}$  y  $\boldsymbol{\omega}_{ck}$ , los parámetros de movimiento dentro de cada región pueden obtenerse mediante mínimos cuadrados lineales.

Si  $p_k$  es el número de puntos de la imagen contenidos en la región  $k$  ( $k = 1, 2, \dots, N$ ), se define un vector fila de seis componentes  $\mathbf{b}_k(\mathbf{x}_j)$  asociado a cada punto  $\mathbf{x}_j$  ( $j = 1, 2, \dots, p_k$ ) de la imagen, mostrado en la ecuación (4.16), donde  $s_i$  y  $q_i$  representan la componente  $i$  de los vectores  $\mathbf{s}$  y  $\mathbf{q}$  respectivamente (mostrados en la ecuación (4.13)).

$$\mathbf{b}_k(\mathbf{x}_j) = \left( \frac{s_1(\mathbf{x}_j)}{Z_c(\mathbf{x}_j)}, \frac{s_2(\mathbf{x}_j)}{Z_c(\mathbf{x}_j)}, \frac{s_3(\mathbf{x}_j)}{Z_c(\mathbf{x}_j)}, q_1(\mathbf{x}_j), q_2(\mathbf{x}_j), q_3(\mathbf{x}_j) \right) \quad (4.16)$$

Las componentes de velocidad lineal  $\mathbf{v}_{ck}$  y angular  $\boldsymbol{\omega}_{ck}$  pueden obtenerse resolviendo un sistema de ecuaciones lineales como el mostrado en la ecuación (4.17):

$$\mathbf{B}_k \boldsymbol{\rho}_k = \mathbf{c}_k \quad k = 1, \dots, N \quad (4.17)$$

donde:  $\boldsymbol{\rho}_k = (\mathbf{v}_{ck}, \boldsymbol{\omega}_{ck})^T$  es el vector que representa las seis componentes del movimiento rígido correspondientes a la región  $\Omega_k$ . Si se entiende como región  $\{\Omega_k\}_{k=1}^{N-1}$  a todos los píxeles contenidos dentro de la curva  $\{\gamma_k\}_{k=1}^{N-1}$  y como región  $\Omega_N$  a los píxeles correspondientes al fondo de la escena. Los vectores  $\mathbf{B}_k$  y  $\mathbf{c}_k$  se definen como:

$$\mathbf{B}_k = \begin{pmatrix} \mathbf{b}_k(x_1) \\ \vdots \\ \mathbf{b}_k(x_{p_k}) \end{pmatrix} \quad y \quad \mathbf{c}_k = \begin{pmatrix} -I_t(x_1) \\ \vdots \\ -I_t(x_{p_k}) \end{pmatrix} \quad k = 1, \dots, N$$

### Estimación de la profundidad mediante descenso por el gradiente

En la segunda etapa la función a minimizar para recuperar la profundidad es la mostrada en la ecuación (4.18) en la que:  $\chi_k$  es la función característica de la región  $\Omega_k$ .

$$E(Z) = \sum_{k=1}^N \int_{\Omega_k} [\psi_k^2(\mathbf{x}) + \mu g(\|\nabla Z\|)] d\mathbf{x} = \int_{\Omega} \sum_{k=1}^N [\chi_k(\mathbf{x}) (\psi_k^2(\mathbf{x}) + \mu g(\|\nabla Z\|))] d\mathbf{x} \quad (4.18)$$

Para obtener la derivada de la función objetivo (mostrada en la ecuación (4.14)) respecto de  $Z$ , dicha función objetivo puede escribirse de forma general como se muestra en la ecuación (4.19). En esta ecuación  $Z_x$  y  $Z_y$  representan respectivamente la

componente  $x$  e  $y$  del gradiente de la profundidad  $\nabla Z = (Z_x, Z_y)$  y  $L(x, y, Z, Z_x, Z_y) = \sum_{k=1}^N \chi_k (\psi_k^2 + \mu g(\|\nabla Z\|))$ .

$$E(Z) = \int_{\Omega} L(x, y, Z, Z_x, Z_y) \quad (4.19)$$

La derivada de  $E$  respecto de  $Z$  es la mostrada en (4.20):

$$\frac{\partial E}{\partial Z} = \frac{\partial L}{\partial Z} - \frac{\partial}{\partial x} \left( \frac{\partial L}{\partial Z_x} \right) - \frac{\partial}{\partial y} \left( \frac{\partial L}{\partial Z_y} \right) \quad (4.20)$$

siendo las derivadas parciales:

$$\frac{\partial L}{\partial Z} = \sum_{k=1}^N \chi_k \left( -2 \frac{\mathbf{s} \cdot \mathbf{v}_{ck}}{Z^2} \right) \psi_k \quad (4.21)$$

$$\frac{\partial L}{\partial Z_x} = \sum_{k=1}^N \chi_k \left( \mu g'(\|\nabla Z\|) \frac{Z_x}{\|\nabla Z\|} \right) \quad (4.22)$$

$$\frac{\partial L}{\partial Z_y} = \sum_{k=1}^N \chi_k \left( \mu g'(\|\nabla Z\|) \frac{Z_y}{\|\nabla Z\|} \right) \quad (4.23)$$

Sustituyendo las expresiones de las derivadas parciales: (4.21), (4.22) y (4.23) en la ecuación (4.20) se obtiene la siguiente expresión para la derivada de la función objetivo (ecuación (4.14)) respecto de la profundidad  $Z$ .

$$\frac{\partial E}{\partial Z} = \sum_{k=1}^N \chi_k \left[ -2 \frac{\mathbf{s} \cdot \mathbf{v}_{ck}}{Z^2} \psi_k - \mu \left( \frac{\partial}{\partial x} \left( g'(\|\nabla Z\|) \frac{Z_x}{\|\nabla Z\|} \right) + \frac{\partial}{\partial y} \left( g'(\|\nabla Z\|) \frac{Z_y}{\|\nabla Z\|} \right) \right) \right] \quad (4.24)$$

La expresión (4.24) puede reescribirse de forma que la derivada de (4.18) con respecto a  $Z$  es:

$$\frac{\partial E}{\partial Z} = \sum_{k=1}^N \chi_k \left[ -2 \frac{\mathbf{s} \cdot \mathbf{v}_{ck}}{Z^2} \psi_k - \mu \operatorname{div} \left( \frac{g'(\|\nabla Z\|)}{\|\nabla Z\|} \nabla Z \right) \right] \quad (4.25)$$

Dado el conjunto de curvas  $\{\gamma_k\}_{k=1}^{N-1}$  que dividen la imagen en  $N$  regiones  $\{\Omega_k\}_{k=1}^N$ , a partir de la ecuación (4.25) se obtienen las derivadas para cada una de las  $N$  regiones  $\Omega_k$  (definidas en la ecuación (4.26)) a las que se añade la condición de contorno de Neumann  $\partial Z / \partial \mathbf{n} = 0$ , siendo  $\mathbf{n}$  el vector unitario normal al borde de la región  $\Omega_k$ .

$$\frac{\partial E}{\partial Z} = -2 \frac{\mathbf{s} \cdot \mathbf{v}_{ck}}{Z^2} \psi_k - \mu \operatorname{div} \left( \frac{g'(\|\nabla Z\|)}{\|\nabla Z\|} \nabla Z \right) \quad k = 1, \dots, N \quad (4.26)$$

Las ecuaciones de descenso correspondientes a (4.26), para cada una de las  $N$  regiones en que se ha dividido la imagen, son las siguientes:

$$\frac{\partial Z}{\partial \tau}(\mathbf{x}, \tau) = 2 \frac{\mathbf{s} \cdot \mathbf{v}_{ck}}{Z^2} \psi_k + \mu \operatorname{div} \left( \frac{g'(\|\nabla Z\|)}{\|\nabla Z\|} \nabla Z \right) \quad k = 1, \dots, N \quad (4.27)$$

donde  $\tau$  indica el tiempo de ejecución del algoritmo y  $g'$  es la derivada ordinaria de la función de preservación de bordes  $g$ .

Existen diferentes alternativas a la hora de elegir la función de preservación de bordes ( $g(s)$ ). En el trabajo de [Sekkati & Mitiche 2006b] se propone la utilización de una función cuadrática ( $g(s) = s^2$ ) sin embargo, en esta tesis se ha decidido realizar un estudio más completo en el que se han considerado diferentes funciones de preservación de bordes. Las diferentes funciones de bordes consideradas se describen con mayor detalle en el capítulo 7. En cualquier caso es necesario sustituir  $g'(\|\nabla Z\|)$  por la derivada de la función elegida, particularizada para el  $\|\nabla Z\|$  en la ecuación (4.27) y posteriormente desarrollar la expresión de la divergencia que aparece en dicha ecuación para obtener la expresión final de la ecuación de descenso.

### Evolución de las curvas para segmentación de movimiento 3D

Dada la profundidad  $Z$  y los parámetros del movimiento rígido 3D  $(\mathbf{v}_k, \boldsymbol{\omega}_k)_{k=1}^N$  (las componentes de velocidad lineal y angular en  $\Gamma_c$ ) en cada una de las  $N$  regiones en las que se ha dividido la imagen, la función objetivo a minimizar con respecto a las curvas que definen los contornos de los robots móviles se muestra en la siguiente ecuación, donde  $\xi_k(\mathbf{x}) = \psi_k^2(\mathbf{x}) + \mu g(\|\nabla Z\|)$ :

$$E[\{\gamma_k\}_{k=1}^{N-1}] = \sum_{k=1}^N \int_{\Omega_k} \xi_k(\mathbf{x}) d\mathbf{x} + \lambda \sum_{k=1}^{N-1} \oint_{\gamma_k} ds \quad (4.28)$$

Para la segmentación de múltiples regiones se obtienen las siguientes derivadas:

$$\frac{\partial E_k}{\partial \gamma_k} = (\xi_k - \varphi_k + \lambda \kappa_{\gamma_k}) \mathbf{n}_k, \quad k = 1, \dots, N-1 \quad (4.29)$$

y las correspondientes ecuaciones de descenso de *Euler-Lagrange*:

$$\frac{\partial \gamma_k}{\partial \tau}(s, \tau) = -(\xi_k(\gamma_k(s)) - \varphi_k(\gamma_k(s)) + \lambda \kappa_{\gamma_k}(\gamma_k(s))) \times \mathbf{n}_k(\gamma_k(s)), \quad k = 1, \dots, N-1 \quad (4.30)$$

donde  $\kappa_{\gamma_k}$  es la curvatura media,  $\mathbf{n}_k$  es la función unitaria normal al exterior de la curva  $\gamma_k$  y las funciones  $\varphi_k$  se definen como:

$$\varphi_k(\gamma_k(s)) = \min_{i \neq k} \xi_i(\gamma_k(s)) \quad (4.31)$$

Para la implementación se representa cada curva mediante el conjunto de nivel cero de una función  $\Phi_k$ , de forma que el interior de cada curva  $\gamma_k$  se corresponde con el conjunto de puntos en los que  $\Phi_k$  es mayor que cero ( $\{\Phi_k > 0\}$ ), mientras que el exterior de la citada curva se corresponde con los puntos para los que la función  $\Phi_k$  es inferior a cero. Obteniéndose, según se describe en [Sekkati & Mitiche 2006b], el siguiente sistema de ecuaciones diferenciales parciales acopladas:

$$\frac{\partial \Phi_k}{\partial \tau}(\mathbf{x}, \tau) = -(\xi_k(\mathbf{x}) - \varphi_k(\mathbf{x}) + \lambda \kappa_{\Phi_k}(\mathbf{x})) \|\nabla \Phi_k(\mathbf{x})\| \quad k = 1, \dots, N-1 \quad (4.32)$$

En las expresiones de la ecuación (4.32), la curvatura media  $\kappa_{\Phi_k}$  se define como:  $\text{div}(\nabla \Phi_k / \|\nabla \Phi_k\|)$  y la función  $\varphi_k(\mathbf{x})$  es la mostrada en la ecuación (4.33):

$$\varphi_k(\mathbf{x}) = \begin{cases} \min \xi_i(\mathbf{x}) & \text{for } i \neq k \in \{1, \dots, N-1\}, \Phi_i(\mathbf{x}) > 0 \\ \xi_N(\mathbf{x}) & \text{else} \end{cases} \quad (4.33)$$

Las tres etapas descritas se repiten hasta que el algoritmo de minimización alcanza la convergencia. Dado que la función objetivo de la ecuación (4.14) es positiva, y decrece con cada iteración, el proceso de minimización converge a un mínimo local. Tanto el número de iteraciones necesarias hasta la convergencia, como la calidad final de la segmentación dependen del criterio de convergencia fijado, la inicialización de las curvas, las constantes de ponderación de cada uno de los términos de la función objetivo ( $\lambda$  y  $\mu$ ) y la función de preservación de bordes ( $g(a)$ ) elegida.

#### 4.1.3.2. Criterio de convergencia

Como ya se ha comentado anteriormente, se considera que el algoritmo de minimización de energía converge cuando las variables implicadas en el mismo dejan de evolucionar de forma significativa entre una iteración y la siguiente. Esto nos permite establecer una condición de parada del algoritmo.

Partiendo de que la función objetivo (ecuación (4.14)) depende de tres grupos de parámetros, es necesario estudiar la evolución de cada uno de estos grupos de forma independiente. Además, se considera que el algoritmo ha alcanzado la convergencia únicamente si la variación de todos los grupos de variables es suficientemente pequeña.

Tras observar la evolución de las variables para diferentes imágenes de test, se han establecido los siguientes valores para cada uno de los grupos de variables incluidos en la función objetivo presentada en la ecuación (4.14):

- En el caso de las componentes de velocidad 3D ( $\mathbf{v}_c$  y  $\boldsymbol{\omega}_c$ ) el criterio de convergencia establecido considera que la variación de velocidades converge cuando la media de la diferencia de velocidades entre una iteración y la siguiente es inferior a 0.5milímetros/segundo, en el caso de la velocidad lineal, y de 0.5grados/segundo para la angular.
- Con respecto a la profundidad, el criterio de convergencia del descenso por el gradiente se cumple si la variación media de la profundidad en los puntos de la imagen, entre una iteración y la siguiente, es inferior a 10mm.
- Finalmente, se considera que la evolución de las curvas se detiene cuando los contornos que definen las diferentes regiones en que se divide la imagen cambian en menos de tres píxeles entre una iteración y la siguiente.

Estos valores se muestran de forma resumida en la Tabla 4.1. En esta tabla se denomina criterio de convergencia a la variación máxima que puede tener una variable entre una iteración y la anterior para que se considere que la citada variable converge.

Grupo de Variables	Criterio de convergencia
Velocidad lineal: $\{\mathbf{v}_{ck}\}_{k=1}^N$	Variación inferior a 0.5 milímetros
Velocidad angular: $\{\boldsymbol{\omega}_{ck}\}_{k=1}^N$	Variación inferior a 0.5 grados
Profundidad: $Z_c$	Variación inferior a 10 milímetros
Curvas: $\{\gamma_k\}_{k=1}^{N-1}$	Variación inferior a 3 píxeles

Tabla 4.1. Criterio de convergencia: valor máximo de variación de la variable entre una iteración y la siguiente para el que se considera que la variable converge. Se indican los diferentes grupos de variables implicados en la función objetivo de la ecuación (4.14).

#### **4.1.4. Fusión de los resultados obtenidos para múltiples cámaras**

Tras la minimización de la función objetivo definida en la ecuación (4.14), es posible obtener las curvas que definen los contornos de los robots móviles sobre el plano imagen. Así mismo se obtienen los parámetros de movimiento 3D (componentes de velocidad lineal y angular) respecto al sistema de referencia de la cámara, y la posición 3D de dichos robots, a falta de un factor de escala debido al uso de una única cámara.

Dado que en el espacio inteligente se cuenta con múltiples cámaras, se propone la incorporación de la información adquirida por todas las cámaras disponibles al resultado final. Esto se hace con el objetivo de incrementar la robustez del proceso de segmentación y posicionamiento 3D. En esta línea, la solución más inmediata pasa por realizar la fusión de los resultados obtenidos tras ejecutar el proceso de minimización de la función objetivo presentada en el apartado 4.1 para cada una de las cámaras disponibles. Con el uso de múltiples cámaras el diagrama de bloques del sistema propuesto queda tal como se muestra en la Figura 4.2

En el diagrama de bloques de la Figura 4.2 se presentan las principales etapas del proceso de segmentación de movimiento y posicionamiento 3D utilizando la función objetivo propuesta en [Sekkti & Mitiche 2006b] (definida en la ecuación (4.14) de este capítulo). En esta figura se puede observar como, tanto la inicialización de las curvas y la profundidad, como el algoritmo de segmentación y posicionamiento 3D se ejecutan, de forma independiente, para cada una de las cámaras disponibles. Posteriormente, se incluye un proceso de fusión de los resultados obtenidos. Cabe destacar que, durante el arranque del sistema, así como cada vez que se detecta la entrada de un nuevo robot, se ejecuta un proceso de inicialización “global”. En este proceso se incluye la información de todas las cámaras disponibles para la estimación del valor inicial de la profundidad para cada uno de los robots móviles presentes en la escena.

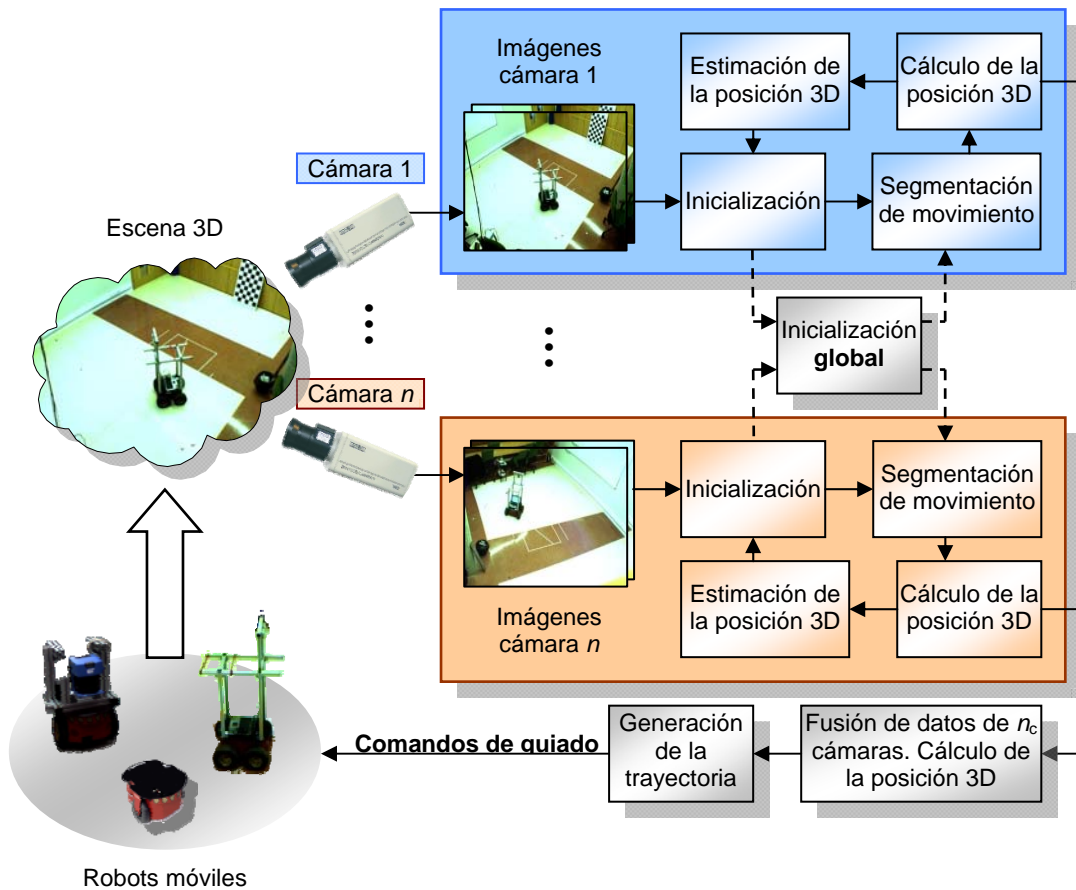


Figura 4.2. Diagrama de bloques general del proceso de segmentación de movimiento y obtención de la posición 3D de robots móviles a partir de las imágenes captadas por un conjunto de cámaras ubicadas en posiciones fijas en el espacio.

Como ya se ha comentado, el objetivo que se persigue con la etapa de fusión de resultados de  $n_c$  cámaras es el incremento de la robustez del sistema. Esto se logra al incluir la información de múltiples cámaras para la obtención de la segmentación de movimiento y la posición 3D de los robots móviles. Sin embargo, dado que los resultados arrojados tras la minimización de la función objetivo (ecuación (4.14)) están referidos al sistema de coordenadas de cada cámara, de forma previa a la fusión, es necesario expresar dichos resultados con respecto a un sistema de referencia común. En este caso, la elección más inmediata es el sistema de referencia del mundo  $\Gamma_w$  que es común a todas las cámaras. Para expresar la relación de los resultados entre los diferentes sistemas de coordenadas se emplearán las matrices transformación  $\{\mathbf{R}_{w c_i}, \mathbf{t}_{w c_i}\}_{i=1}^{n_c}$  entre el sistema de referencia de cada cámara  $\{\Gamma_{c_i}\}_{i=1}^{n_c}$  y el del mundo  $\Gamma_w$ .

Tras expresar los resultados de la minimización de la función objetivo con relación a  $\Gamma_w$  (componentes de la velocidad lineal y angular, y posición 3D del centroide de los robots móviles), el resultado global de cada parámetro se calcula hallando el valor medio del resultado de dicho parámetro en todas las cámaras. Es decir, si

$\{\mathbf{v}_{wi,k} = (v_{wi,k}^x, v_{wi,k}^y, v_{wi,k}^z), \boldsymbol{\omega}_{wi,k} = (\omega_{wi,k}^x, \omega_{wi,k}^y, \omega_{wi,k}^z)\}_{i=1:n_c}^{k=1:N}$  son las componentes de la velocidad lineal y angular del objeto  $k$  obtenidas tras la minimización de la función objetivo en cada una de las cámaras ( $i=1,2,\dots,n_c$ ), el resultado de la fusión de resultados viene dado por las expresiones mostradas en (4.34).

$$\begin{aligned} \mathbf{v}_{wk} &= \frac{1}{n_c} \sum_{i=1}^{n_c} \mathbf{v}_{wi,k} \\ \boldsymbol{\omega}_{wk} &= \frac{1}{n_c} \sum_{i=1}^{n_c} \boldsymbol{\omega}_{wi,k} \end{aligned} \quad k = 1, 2, \dots, N \quad (4.34)$$

Del mismo modo, si identificamos como  $\{\mathbf{P}_{wi,k} = (X_{wi,k}, Y_{wi,k}, Z_{wi,k})\}_{i=1:n_c}^{k=1:N}$  las coordenadas de los centroides de los robots móviles estimados para la cámara  $k$ , y expresados con respecto al sistema de referencia  $\Gamma_w$ , el valor global de dichos centroides se obtiene mediante la ecuación (4.35)

$$\mathbf{P}_{wk} = \frac{1}{n_c} \sum_{i=1}^{n_c} \mathbf{P}_{wi,k}, \quad k = 1, 2, \dots, N \quad (4.35)$$

El uso de la media aritmética permite realizar la fusión de resultados de forma sencilla, obteniendo una ligera mejora en la fiabilidad y la robustez del sistema al incorporar la información de múltiples cámaras al resultado de la segmentación y posicionamiento 3D. Sin embargo, aunque esta técnica presenta algunas ventajas, debidas a la independencia entre las distintas cámaras (permite realizar la minimización de la función objetivo en un equipo diferente para cada una de las cámaras), también presenta desventajas. Una de las desventajas es que los resultados obtenidos son poco robustos, a pesar del proceso de fusión posterior a la minimización, debido a que en la minimización únicamente se tiene en cuenta la información de una cámara. Por este motivo, en esta tesis se propone una nueva función objetivo que incorpora la información adquirida por múltiples cámaras. Esta nueva función se describe con detalle a continuación, en el apartado 4.2.



## 4.2. Segmentación de movimiento y posicionamiento 3D mediante la minimización de una función objetivo definida para múltiples cámaras

Ya se han comentado los problemas derivados del hecho de considerar la información adquirida por una única cámara en el proceso de minimización de la función objetivo (ecuación (4.14)). Para tratar de solventar estos problemas, en esta tesis se propone una nueva función objetivo en la que se incorpora la información adquirida por las  $n_c$  cámaras disponibles en el espacio inteligente. Con objeto de relacionar la información de movimiento en cada una de las cámaras ha sido necesario definir de nuevo la restricción de brillo 3D, (4.12). Esta nueva definición se realiza en función de las componentes de velocidad lineal y angular en el sistema de referencia global  $\Gamma_w$  en lugar de hacerlo respecto a dichas componentes de velocidad en el sistema de referencia de cada cámara  $\Gamma_{ci}$  ( $i=1,2,\dots, n_c$ ). Esto es así, ya que, en el sistema de referencia del mundo  $\Gamma_w$ , las componentes de velocidad son las mismas para todas las cámaras que ven un robot en un determinado instante.

La nueva restricción de brillo 3D para múltiples cámaras se presenta a continuación, en el apartado 4.2.1. Tras definir esta restricción de brillo 3D es sencillo generalizar la función objetivo de la ecuación (4.14) para incorporar en ella la información de las  $n_c$  cámaras, según se describe en el apartado 4.2.2. El hecho de incluir la información de múltiples cámaras en el proceso de minimización de la función objetivo permite incrementar de forma notable la robustez del sistema, tal como se verá en el capítulo 9 en el que se presentan las pruebas realizadas para la validación de la solución propuesta.

Al igual que en el caso de la función objetivo descrita en el apartado 4.1, de forma previa a la minimización es necesario inicializar las curvas que definen los contornos de los posibles robots móviles presentes en la escena, y la profundidad (coordenada  $Z_{ci}$  en el sistema de referencia de cada cámara  $\Gamma_{ci}$ ,  $i=1,2,\dots, n_c$ ) para cada uno de los píxeles. Dada la importancia de la correcta inicialización de estos parámetros, este proceso se describe con detalle a lo largo de los capítulos 5 y 6 de esta tesis.

Tras la inicialización de variables, la minimización de la función objetivo se lleva a cabo mediante un algoritmo *greedy*, e iterativo en tres etapas que se detallan a lo largo del apartado 4.2.3

Las diferentes etapas del proceso de segmentación de movimiento y obtención de la posición 3D mediante minimización de una función objetivo para  $n_c$  cámaras se muestran en el diagrama de bloques de la Figura 4.3.

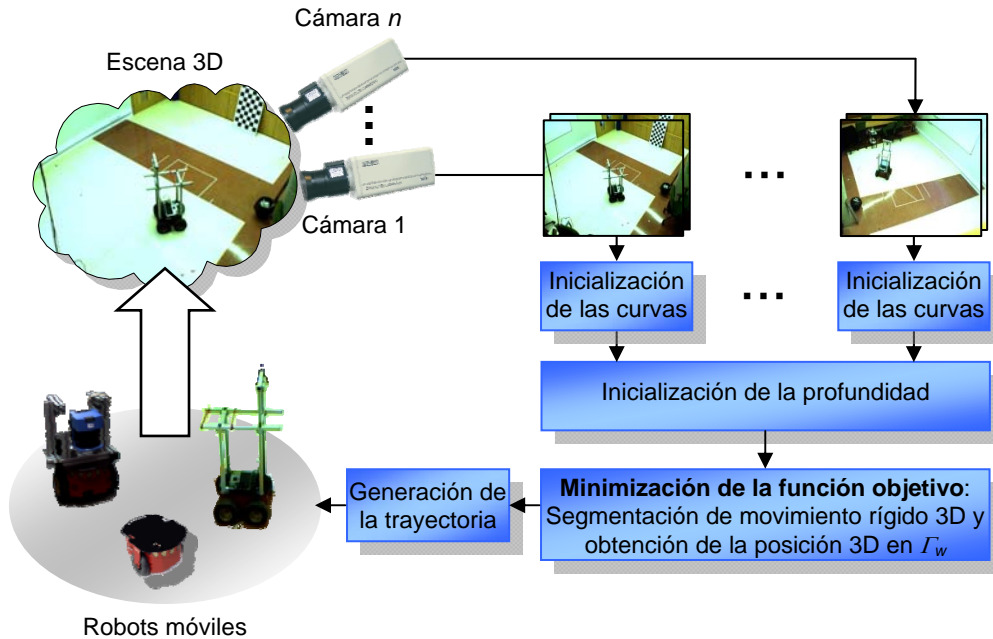


Figura 4.3. Diagrama de bloques general del algoritmo de segmentación de movimiento rígido y obtención de la posición 3D a partir de las imágenes captadas por  $n_c$  cámaras ubicadas en posiciones fijas del espacio. Tanto la segmentación de movimiento como la obtención de la posición 3D se realizan de forma conjunta para las  $n_c$  imágenes, mientras que el proceso de inicialización de las curvas se ejecuta por cada una de las cámaras disponibles.

A continuación, se presenta la función objetivo para  $n_c$  cámaras propuesta en esta tesis. En el apartado 4.2.1 se describe la nueva restricción de brillo 3D para, posteriormente, en el apartado 4.2.2 mostrar la función objetivo en la que se incorpora la información adquirida por las  $n_c$  cámaras fijas del espacio inteligente.

#### 4.2.1. Restricción de brillo 3D para $n_c$ cámaras

Si se identifican como  $\mathbf{P}_w = (X_w, Y_w, Z_w)^T$  las coordenadas 3D de un punto  $\mathbf{P}$  perteneciente a un sólido rígido en movimiento en el sistema de referencia global  $\Gamma_w$ , la velocidad de  $\mathbf{P}_w$  en  $\Gamma_w$  puede expresarse mediante la ecuación (4.36) donde  $\mathbf{v}_w$  y  $\boldsymbol{\omega}_w$  son las componentes de velocidad lineal y angular del punto  $\mathbf{P}$  referidas a  $\Gamma_w$ .

$$\dot{\mathbf{P}}_w = \begin{pmatrix} \dot{X}_w & \dot{Y}_w & \dot{Z}_w \end{pmatrix}^T = \mathbf{v}_w + \boldsymbol{\omega}_w \times \mathbf{P}_w \quad (4.36)$$

Si denominamos  $\mathbf{R}_{wc}$  a la matriz de rotación y  $\mathbf{t}_{wc}$  al vector de traslación entre el sistema de referencia global  $\Gamma_w$  y el de la cámara  $\Gamma_c$ , las coordenadas de  $\mathbf{P}$  referidas a  $\Gamma_c$  pueden expresarse en función de las coordenadas en  $\Gamma_w$  usando la ecuación (4.37).

$$\mathbf{P}_c = \mathbf{R}_{wc} \mathbf{P}_w + \mathbf{t}_{wc} \quad (4.37)$$

Derivando la expresión (4.37) respecto del tiempo se obtiene la velocidad de  $\mathbf{P}$  referida a  $\Gamma_c$  en función de la velocidad expresada respecto a  $\Gamma_w$ .

$$\dot{\mathbf{P}}_c = \mathbf{R}_{wc} \dot{\mathbf{P}}_w \quad (4.38)$$

Sustituyendo  $\dot{\mathbf{P}}_c$  y  $\dot{\mathbf{P}}_w$  en (4.38) por sus expresiones (mostradas en las ecuaciones (4.3) y (4.36) respectivamente) se obtiene la siguiente igualdad:

$$\mathbf{v}_c + \boldsymbol{\omega}_c \times \mathbf{P}_c = \mathbf{R}_{wc} (\mathbf{v}_w + \boldsymbol{\omega}_w \times \mathbf{P}_w) \quad (4.39)$$

Teniendo en cuenta que los productos vectoriales ( $\boldsymbol{\omega} \times \mathbf{P}$ ) pueden sustituirse por productos escalares  $\hat{\boldsymbol{\omega}} \cdot \mathbf{P}$ , siendo  $\hat{\boldsymbol{\omega}}$  la matriz antisimétrica:

$$\hat{\boldsymbol{\omega}} = \begin{pmatrix} 0 & -\omega^z & \omega^y \\ \omega^z & 0 & -\omega^x \\ -\omega^y & \omega^x & 0 \end{pmatrix}$$

La ecuación (4.38) puede reescribirse como:

$$\mathbf{R}_{wc} (\mathbf{v}_w + \hat{\boldsymbol{\omega}}_w \cdot \mathbf{P}_w) = \mathbf{v}_c + \hat{\boldsymbol{\omega}}_c \cdot \mathbf{P}_c \quad (4.40)$$

Sustituyendo además  $\mathbf{P}_w$  por su expresión en función de las coordenadas de  $\mathbf{P}$  en  $\Gamma_c$  ( $\mathbf{P}_w = \mathbf{R}_{wc}^T (\mathbf{P}_c - \mathbf{t}_{wc})$ ) se obtiene:

$$\mathbf{R}_{wc} \mathbf{v}_w - \mathbf{R}_{wc} \hat{\boldsymbol{\omega}}_w \cdot \mathbf{R}_{wc}^T \mathbf{t}_{wc} + \mathbf{R}_{wc} \hat{\boldsymbol{\omega}}_w \cdot \mathbf{R}_{wc}^T \mathbf{P}_c = \mathbf{v}_c + \hat{\boldsymbol{\omega}}_c \cdot \mathbf{P}_c \quad (4.41)$$

Por tanto, las componentes de velocidad lineal y angular en el sistema de referencia  $\Gamma_c$  pueden expresarse en función de las componentes de velocidad en  $\Gamma_w$  utilizando las siguientes ecuaciones:

$$\begin{aligned} \mathbf{v}_c &= \mathbf{R}_{wc} \mathbf{v}_w - \mathbf{R}_{wc} \hat{\boldsymbol{\omega}}_w \cdot \mathbf{R}_{wc}^T \mathbf{t}_{wc} \\ \hat{\boldsymbol{\omega}}_c &= \mathbf{R}_{wc} \hat{\boldsymbol{\omega}}_w \cdot \mathbf{R}_{wc}^T \end{aligned} \quad (4.42)$$

Si se desarrolla la expresión de  $\hat{\boldsymbol{\omega}}_c$  suponiendo que la matriz de rotación  $\mathbf{R}_{wc}$  puede expresarse según se muestra en la ecuación (4.43), se obtienen las expresiones

mostradas en la ecuación (4.44) para cada una de las componentes de la velocidad angular respecto al sistema de referencia de la cámara ( $\Gamma_c$ )

$$\mathbf{R}_{wc} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \begin{pmatrix} \mathbf{R}_{wc}^1 \\ \mathbf{R}_{wc}^2 \\ \mathbf{R}_{wc}^3 \end{pmatrix} \quad (4.43)$$

$$\begin{aligned} \omega_c^x &= (r_{22}r_{33} - r_{23}r_{32})\omega_w^x + (r_{23}r_{31} - r_{21}r_{33})\omega_w^y + (r_{21}r_{32} - r_{22}r_{31})\omega_w^z \\ \omega_c^y &= (r_{13}r_{32} - r_{12}r_{33})\omega_w^x + (r_{11}r_{33} - r_{13}r_{21})\omega_w^y + (r_{12}r_{31} - r_{11}r_{32})\omega_w^z \\ \omega_c^z &= (r_{12}r_{23} - r_{13}r_{22})\omega_w^x + (r_{13}r_{21} - r_{11}r_{23})\omega_w^y + (r_{11}r_{22} - r_{12}r_{21})\omega_w^z \end{aligned} \quad (4.44)$$

Estas ecuaciones se pueden escribir en función de la matriz adjunta de  $\mathbf{R}_{wc}$  utilizando la expresión mostrada en la ecuación (4.45).

$$\begin{pmatrix} \omega_c^x \\ \omega_c^y \\ \omega_c^z \end{pmatrix} = \text{adj}(\mathbf{R}_{wc}) \begin{pmatrix} \omega_w^x \\ \omega_w^y \\ \omega_w^z \end{pmatrix} \quad (4.45)$$

Quedando la ecuación (4.41) como sigue:

$$\dot{\mathbf{P}}_c = \begin{pmatrix} \dot{X}_c & \dot{Y}_c & \dot{Z}_c \end{pmatrix}^T = \mathbf{R}_{wc} \mathbf{v}_w + \text{adj}(\mathbf{R}_{wc}) \omega_w \times (\mathbf{P}_w - \mathbf{t}_{wc}) \quad (4.46)$$

A partir de la ecuación (4.46) es posible obtener las expresiones de cada una de las componentes de la velocidad del punto  $\mathbf{P}$  en  $\Gamma_c$ :  $(\dot{X}_c, \dot{Y}_c, \dot{Z}_c)^T$  en función de las velocidades lineal y angular en  $\Gamma_w$  y de las matrices de transformación entre ambos sistemas de referencia ( $\mathbf{R}_{wc}$ ,  $\mathbf{T}_{wc}$ ).

Denominando  $(t_{wc}^x \ t_{wc}^y \ t_{wc}^z)^T$  a las componentes del vector de traslación  $\mathbf{t}_{wc}$  y considerando que la matriz adjunta de  $\mathbf{R}_{wc}$  puede expresarse como:

$$\text{adj}(\mathbf{R}_{wc}) = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

se obtienen las siguientes expresiones de componentes de velocidad de  $\mathbf{P}$  en  $\Gamma_c$  en función de las velocidades en  $\Gamma_w$ :

$$\begin{aligned} \dot{X}_c &= r_{11}v_w^x + r_{12}v_w^y + r_{13}v_w^z + [a_{21}(Z_c - t_{wc}^z) - a_{31}(Y_c - t_{wc}^y)]\omega_w^x \\ &\quad + [a_{22}(Z_c - t_{wc}^z) - a_{32}(Y_c - t_{wc}^y)]\omega_w^y + [a_{23}(Z_c - t_{wc}^z) - a_{33}(Y_c - t_{wc}^y)]\omega_w^z \end{aligned} \quad (4.47)$$

$$\begin{aligned} \dot{Y}_c = & r_{21}v_w^x + r_{22}v_w^y + r_{23}v_w^z + [a_{31}(X_c - t_{wc}^x) - a_{11}(Z_c - t_{wc}^z)]\omega_w^x \\ & + [a_{32}(X_c - t_{wc}^x) - a_{12}(Z_c - t_{wc}^z)]\omega_w^y + [a_{33}(X_c - t_{wc}^x) - a_{13}(Z_c - t_{wc}^z)]\omega_w^z \end{aligned} \quad (4.48)$$

$$\begin{aligned} \dot{Z}_c = & r_{31}v_w^x + r_{32}v_w^y + r_{33}v_w^z + [a_{11}(Y_c - t_{wc}^y) - a_{21}(X_c - t_{wc}^x)]\omega_w^x \\ & + [a_{12}(Y_c - t_{wc}^y) - a_{22}(X_c - t_{wc}^x)]\omega_w^y + [a_{13}(Y_c - t_{wc}^y) - a_{23}(X_c - t_{wc}^x)]\omega_w^z \end{aligned} \quad (4.49)$$

Sustituyendo  $X_c$  e  $Y_c$  por las funciones de  $(x, y)$  definidas en la ecuación (4.8), las ecuaciones anteriores se transforman en:

$$\begin{aligned} \dot{X}_c = & r_{11}v_w^x + r_{12}v_w^y + r_{13}v_w^z \\ & + \left[ \left( a_{21} - \frac{y-s_2}{f_y} a_{31} \right) Z_c + a_{31}t_{wc}^y - a_{21}t_{wc}^z \right] \omega_w^x \\ & + \left[ \left( a_{22} - \frac{y-s_2}{f_y} a_{32} \right) Z_c + a_{32}t_{wc}^y - a_{22}t_{wc}^z \right] \omega_w^y \\ & + \left[ \left( a_{23} - \frac{y-s_2}{f_y} a_{33} \right) Z_c + a_{33}t_{wc}^y - a_{23}t_{wc}^z \right] \omega_w^z \end{aligned} \quad (4.50)$$

$$\begin{aligned} \dot{Y}_c = & r_{21}v_w^x + r_{22}v_w^y + r_{23}v_w^z \\ & + \left[ \left( \frac{x-s_1}{f_x} a_{31} - a_{11} \right) Z_c + a_{11}t_{wc}^z - a_{31}t_{wc}^x \right] \omega_w^x \\ & + \left[ \left( \frac{x-s_1}{f_x} a_{32} - a_{12} \right) Z_c + a_{12}t_{wc}^z - a_{32}t_{wc}^x \right] \omega_w^y \\ & + \left[ \left( \frac{x-s_1}{f_x} a_{33} - a_{13} \right) Z_c + a_{13}t_{wc}^z - a_{33}t_{wc}^x \right] \omega_w^z \end{aligned} \quad (4.51)$$

$$\begin{aligned} \dot{Z}_c = & r_{31}v_w^x + r_{32}v_w^y + r_{33}v_w^z \\ & + \left[ \left( \frac{y-s_2}{f_y} a_{11} - \frac{x-s_1}{f_x} a_{21} \right) Z_c + a_{21}t_{wc}^x - a_{11}t_{wc}^y \right] \omega_w^x \\ & + \left[ \left( \frac{y-s_2}{f_y} a_{12} - \frac{x-s_1}{f_x} a_{22} \right) Z_c + a_{22}t_{wc}^x - a_{12}t_{wc}^y \right] \omega_w^y \\ & + \left[ \left( \frac{y-s_2}{f_y} a_{13} - \frac{x-s_1}{f_x} a_{23} \right) Z_c + a_{23}t_{wc}^x - a_{13}t_{wc}^y \right] \omega_w^z \end{aligned} \quad (4.52)$$

Introduciendo las ecuaciones (4.50), (4.51) y (4.52) en las expresiones obtenidas para las componentes de velocidad en el plano imagen  $(u, v)$  mostradas en la ecuación (4.6) se obtienen las siguientes ecuaciones:

$$\begin{aligned} u = & \frac{1}{Z_c} \left[ (f_x r_{11} - (x-s_1)r_{31})v_w^x + (f_x r_{12} - (x-s_1)r_{32})v_w^y + (f_x r_{13} - (x-s_1)r_{33})v_w^z \right] \\ & + (x-s_1) \left[ \frac{t_{wc}^y}{Z_c} - \frac{y-s_2}{f_y} \right] \left[ a_{11}\omega_w^x + a_{12}\omega_w^y + a_{13}\omega_w^z \right] \\ & + \left[ f_y + \frac{(x-s_1)^2}{f_x} - \frac{1}{Z_c} (f_x t_{wc}^z + (x-s_1)t_{wc}^x) \right] \left[ a_{21}\omega_w^x + a_{22}\omega_w^y + a_{23}\omega_w^z \right] \\ & + f_x \left[ \frac{t_{wc}^y}{Z_c} - \frac{y-s_2}{f_y} \right] \left[ a_{31}\omega_w^x + a_{32}\omega_w^y + a_{33}\omega_w^z \right] \end{aligned} \quad (4.53)$$

$$\begin{aligned} v = & \frac{1}{Z_c} \left[ (f_y r_{21} - (x-s_1)r_{31})v_w^x + (f_y r_{22} - (x-s_1)r_{32})v_w^y + (f_y r_{23} - (x-s_1)r_{33})v_w^z \right] \\ & - \left[ f_y + \frac{(y-s_2)^2}{f_y} - \frac{1}{Z_c} (f_y t_{wc}^z + (y-s_2)t_{wc}^y) \right] \left[ a_{11}\omega_w^x + a_{12}\omega_w^y + a_{13}\omega_w^z \right] \\ & - (y-s_2) \left[ \frac{t_{wc}^x}{Z_c} - \frac{x-s_1}{f_x} \right] \left[ a_{21}\omega_w^x + a_{22}\omega_w^y + a_{23}\omega_w^z \right] \\ & - f_y \left[ \frac{t_{wc}^x}{Z_c} - \frac{x-s_1}{f_x} \right] \left[ a_{31}\omega_w^x + a_{32}\omega_w^y + a_{33}\omega_w^z \right] \end{aligned} \quad (4.54)$$

que pueden expresarse en formato matricial:

$$u = \frac{1}{Z_c} (f_x R_{wc}^1 - (x - s_1) R_{wc}^3) \mathbf{v}_w + \mathbf{q}_u \text{adj}(\mathbf{R}_{wc}) \boldsymbol{\omega}_w \quad (4.55)$$

$$v = \frac{1}{Z_c} (f_y R_{wc}^2 - (y - s_2) R_{wc}^3) \mathbf{v}_w + \mathbf{q}_v \text{adj}(\mathbf{R}_{wc}) \boldsymbol{\omega}_w \quad (4.56)$$

donde  $R_{wc}^i$  representa la fila  $i$  de la matriz de rotación entre  $\Gamma_w$  y  $\Gamma_c$  según se ha visto en la ecuación (4.43) y  $\mathbf{q}_u, \mathbf{q}_v$  son los siguientes vectores fila:

$$\mathbf{q}_u = \left( \begin{array}{c} (x - s_1) \left[ \frac{t_{wc}^y}{Z_c} - \frac{y - s_2}{f_y} \right] \\ \left[ f_x + \frac{(x - s_1)^2}{f_x} - \frac{1}{Z_c} (f_x t_{wc}^z + (x - s_1) t_{wc}^x) \right] \\ f_x \left[ \frac{t_{wc}^y}{Z_c} - \frac{y - s_2}{f_y} \right] \end{array} \right)^T \quad (4.57)$$

$$\mathbf{q}_v = - \left( \begin{array}{c} \left[ f_y + \frac{(y - s_2)^2}{f_y} - \frac{1}{Z_c} (f_y t_{wc}^z + (y - s_2) t_{wc}^y) \right] \\ (y - s_2) \left[ \frac{t_{wc}^x}{Z_c} - \frac{x - s_1}{f_x} \right] \\ f_y \left[ \frac{t_{wc}^x}{Z_c} - \frac{x - s_1}{f_x} \right] \end{array} \right)^T \quad (4.58)$$

Sustituyendo las componentes de la velocidad sobre el plano imagen  $u$  y  $v$  en la restricción de brillo mostrada en la ecuación (4.10), se obtiene la siguiente expresión para la restricción de brillo 3D, en función de las velocidades lineal y angular en el sistema de referencia global ( $\mathbf{v}_w$  y  $\boldsymbol{\omega}_w$ )

$$I_t + \mathbf{s} \frac{\mathbf{v}_w}{Z_c} + [\mathbf{q}_u I_x + \mathbf{q}_v I_y] \cdot \text{adj}(\mathbf{R}_{wc}) \boldsymbol{\omega}_w = 0 \quad (4.59)$$

donde  $\mathbf{s}$  es el vector fila de 3 componentes:

$$\mathbf{s} = \left[ f_x I_x \quad f_y I_y \quad -((x - s_1) I_x + (y - s_2) I_y) \right] \quad (4.60)$$

Desarrollando la expresión  $[\mathbf{q}_u I_x + \mathbf{q}_v I_y] \cdot \text{adj}(\mathbf{R}_{wc}) \boldsymbol{\omega}_w$  de la ecuación (4.59) se obtiene:

$$[\mathbf{q}_u I_x + \mathbf{q}_v I_y] \cdot \text{adj}(\mathbf{R}_{wc}) \boldsymbol{\omega}_w = \mathbf{q} \cdot \text{adj}(\mathbf{R}_{wc}) \boldsymbol{\omega}_w + \mathbf{t}_{wc}^T \mathbf{r} \cdot \text{adj}(\mathbf{R}_{wc}) \frac{\boldsymbol{\omega}_w}{Z_c} \quad (4.61)$$

donde  $\mathbf{q}$  es el vector fila de tres componentes mostrado en la ecuación (4.62) y  $\mathbf{r}$  es la matriz antisimétrica presentada en la ecuación (4.63):

$$\mathbf{q} = \left( \begin{array}{c} -f_y \mathbf{I}_y - \frac{(y - s_2)}{f_y} ((x - s_1) \mathbf{I}_x + (y - s_2) \mathbf{I}_y) \\ + f_x \mathbf{I}_x + \frac{(x - s_1)}{f_x} ((x - s_1) \mathbf{I}_x + (y - s_2) \mathbf{I}_y) \\ -\frac{f_x}{f_y} (y - s_2) \mathbf{I}_x + \frac{f_y}{f_x} (x - s_1) \mathbf{I}_y \end{array} \right)^T \quad (4.62)$$

$$\mathbf{r} = \begin{pmatrix} 0 & -((x-s_1)I_x + (y-s_2)I_y) & -f_y I_y \\ ((x-s_1)I_x + (y-s_2)I_y) & 0 & f_x I_x \\ f_y I_y & -f_x I_x & 0 \end{pmatrix} \quad (4.63)$$

Esto permite reescribir la restricción de brillo 3D de la ecuación (4.59) de forma compacta utilizando la siguiente expresión:

$$I_t + \mathbf{s} \cdot \mathbf{R}_{wc} \cdot \frac{\mathbf{v}_w}{Z_c} + \mathbf{q} \cdot \text{adj}(\mathbf{R}_{wc}) \boldsymbol{\omega}_w + \mathbf{t}_{wc}^T \mathbf{r} \cdot \text{adj}(\mathbf{R}_{wc}) \frac{\boldsymbol{\omega}_w}{Z_c} = 0 \quad (4.64)$$

La restricción de brillo 3D mostrada en la ecuación (4.64) debe ser cumplida por cada una de las  $n_c$  cámaras que ven el punto  $\mathbf{P}$ . De esta forma es posible definir una restricción de brillo constante 3D para  $n_c$  cámaras calibradas teniendo en cuenta que las componentes de velocidad de  $\mathbf{P}$  en  $\Gamma_w$  deben ser iguales para todas las cámaras, mientras que  $I_t$  y las matrices  $\mathbf{s}$ ,  $\mathbf{q}$  y  $\mathbf{r}$  dependen de las medidas de cada una de las cámaras. Por otra parte,  $\mathbf{R}_{wc}$  y  $\mathbf{t}_{wc}$  son la matriz de rotación y el vector de traslación entre el sistema de referencia de cada una de las cámaras ( $\Gamma_{ci}$ ) y el sistema de referencia global  $\Gamma_w$ . Por tanto, la restricción de brillo 3D para los píxeles de una región  $k$ , en la cámara  $i$  ( $\psi_{ki}$ ) se define según se muestra en la siguiente ecuación:

$$\psi_{ki}(\mathbf{x}) = I_{ti} + \mathbf{s}_i \cdot \mathbf{R}_{wci} \cdot \frac{\mathbf{v}_{wk}}{Z_{ci}} + \mathbf{q}_i \cdot \text{adj}(\mathbf{R}_{wci}) \boldsymbol{\omega}_{wk} + \mathbf{t}_{wci}^T \mathbf{r}_i \cdot \text{adj}(\mathbf{R}_{wci}) \frac{\boldsymbol{\omega}_{wk}}{Z_{ci}} \quad (4.65)$$

En la ecuación (4.65) se debe destacar que se ha añadido el subíndice  $i$  a todos los vectores y matrices que dependen de cada una de las cámaras. Además, es necesario tener en cuenta que, tanto la derivada temporal  $I_{ti}$ , como las matrices  $\mathbf{q}_i$ ,  $\mathbf{r}_i$  y  $\mathbf{s}_i$ , y la profundidad en el sistema de referencia de la cámara ( $Z_{ci}$ ) son función de las coordenadas de cada píxel en la imagen ( $\mathbf{x} = (x, y)$ ).

## 4.2.2. Función objetivo para $n_c$ cámaras

Del mismo modo que para el caso de una única cámara (presentado en el apartado 4.1.2 de este documento), a partir de la ecuación (4.65) es posible definir una función objetivo para  $n_c$  cámaras en función de tres grupos de parámetros:

- Las curvas  $\{\gamma_{ki}\}_{k=1, \dots, N-1}^{i=1, \dots, n_c}$  que definen los contornos de los  $(N-1)$  robots móviles presentes en las escenas captadas por cada una de las  $n_c$  cámaras,

- Las componentes de la velocidad lineal y angular de cada uno de los  $(N-1)$  robots móviles, así como del fondo de la escena en el sistema de referencia del mundo  $\Gamma_w$ :  $\{\mathbf{v}_{wk}\}_{k=1}^N, \{\boldsymbol{\omega}_{wk}\}_{k=1}^N$  (comunes para todas las cámaras), y
- La distancia de cada punto 3D ( $\mathbf{P}$ ) a cada una de las cámaras ( $i=1, \dots, n_c$ ), que coincide con la coordenada  $Z_{ci}$  de dicho punto en el sistema de referencia de la cámara correspondiente  $\Gamma_{ci}$ .

La función objetivo obtenida para  $n_c$  cámaras se muestra en la ecuación (4.66).

$$\begin{aligned}
 E\left[\{\gamma_{ki}\}_{k=1, \dots, N-1}^{i=1, \dots, n_c}, \{\mathbf{v}_{wk}\}_{k=1}^N, \{\boldsymbol{\omega}_{wk}\}_{k=1}^N, \{Z_{ci}\}_{i=1}^{n_c}\right] = \\
 \sum_{k=1}^N \sum_{i=1}^{n_{ck}} \left[ \int_{\Omega_{ki}} \psi_{ki}^2(\mathbf{x}) d\mathbf{x} + \mu \int_{\Omega_{ki}} g(\|\nabla Z_{ci}\|) d\mathbf{x} \right] + \sum_{k=1}^{N-1} \sum_{i=1}^{n_{ck}} \lambda \oint_{\gamma_{ki}} ds \quad (4.66) \\
 \lambda, \mu \in \mathfrak{R} \text{ y } \lambda, \mu \succ 0
 \end{aligned}$$

En la ecuación (4.66)  $\psi_{ki}$  es la restricción de brillo 3D para los píxeles del robot móvil asociado al contorno de la región  $k$  vista desde la cámara  $i$  (definida en (4.65)).  $\lambda$  y  $\mu$  son constantes reales y positivas que ponderan la contribución de cada uno de los términos y  $\nabla = (\partial_x, \partial_y)$  es el operador gradiente espacial.

Al igual que en el caso de la función objetivo definida para una única cámara, (4.14), en (4.66) la primera integral mide la conformidad de la interpretación 3D de las variaciones espacio-temporales de la secuencia en cada una de las  $N$  regiones a través de la restricción de brillo 3D (definida en (4.66)), mientras que las otras dos integrales de (4.66) son términos de regularización de la profundidad, así como de las  $N-1$  curvas que definen los contornos de las  $N$  regiones en que se divide la imagen.

Si se compara la función objetivo de la ecuación (4.66) con la presentada para una única cámara en la ecuación (4.14), en primer lugar se observa que se ha incluido un segundo sumatorio en el que aparecen las diferentes cámaras. Cabe destacar que por cada una de las  $N-1$  curvas que definen los contornos de los robots móviles se consideran únicamente las  $n_{ck}$  ( $n_{ck} \leq n_c$ ) cámaras en cuyo plano imagen aparece el robot móvil definido por dicho contorno. Esto impone la necesidad de incluir un proceso de identificación de los robots móviles durante la inicialización.



### 4.2.3. Minimización de la función objetivo

De igual manera que en el caso de la minimización de energía para una cámara, descrita en el apartado 4.1.3, para la minimización de la función definida en la ecuación (4.66) se propone un algoritmo *greedy* e iterativo de tres etapas que, tras un proceso de inicialización, se repiten hasta alcanzar la convergencia.

#### 4.2.3.1. Etapas del proceso de minimización

En cada una de las etapas del algoritmo de minimización se suponen fijos y conocidos dos de los tres grupos de parámetros de los que depende la función objetivo y se obtiene el grupo restante para las  $n_c$  cámaras consideradas en cada caso. Cada una de estas etapas se describe a continuación.

#### Obtención de los parámetros del movimiento 3D mediante mínimos cuadrados lineales

En esta primera etapa del algoritmo propuesto para la minimización de la función objetivo mostrada en (4.66) se suponen conocidas:

- las curvas que definen los contornos de los  $N-I$  robots móviles presentes en la escena captada por cada una de las  $n_c$  cámaras,
- el valor de la profundidad  $Z_c$  de cada uno de los puntos proyectados en el plano imagen de cada cámara (coordenada  $Z$  en  $\Gamma_c$ ).

De esta forma, la función objetivo que depende de las componentes de velocidad lineal y angular en el sistema de referencia global ( $\Gamma_w$ ) se reduce a la ecuación (4.67)

$$E\left(\left\{\mathbf{v}_{wk}\right\}_{k=1}^N, \left\{\boldsymbol{\omega}_{wk}\right\}_{k=1}^N\right) = \sum_{k=1}^N \sum_{i=1}^{n_c} \int_{\Omega_{ki}} \psi_{ki}^2(\mathbf{x}) d\mathbf{x} \quad (4.67)$$

Teniendo en cuenta que la restricción de brillo constante para objetos rígidos 3D definida para  $n_c$  cámaras en (4.65) depende linealmente de las componentes de velocidad  $\mathbf{v}_{wk}$  y  $\boldsymbol{\omega}_{wk}$  en el sistema de referencia global  $\Gamma_w$ , los parámetros de movimiento dentro de cada una de las  $N$  regiones en que se ha dividido la imagen de entrada pueden obtenerse mediante mínimos cuadrados, resolviendo un sistema de ecuaciones lineales.

Si se denomina:

$$\begin{aligned} \mathbf{S}_i &= \mathbf{s}_i \cdot \mathbf{R}_{wci} \\ \mathbf{Q}_i &= \mathbf{q}_i \cdot \text{adj}(\mathbf{R}_{wci}) \\ \mathbf{R}_i &= \mathbf{t}_{wci}^T \cdot \mathbf{r}_i \cdot \text{adj}(\mathbf{R}_{wci}) \end{aligned}$$

la restricción de brillo de la ecuación (4.65) puede reescribirse como:

$$\psi_{ki}(\mathbf{x}) = I_{ti} + \mathbf{S}_i \frac{\mathbf{v}_{wk}}{Z_{ci}} + \left( \mathbf{Q}_i + \frac{\mathbf{R}_i}{Z_{ci}} \right) \boldsymbol{\omega}_{wk} \quad (4.68)$$

Por tanto, si  $p_{ki}$  es el número de puntos de la imagen contenidos en la región  $k$  ( $k = 1, 2, \dots, N$ ) vista desde la cámara  $i$  ( $i = 1, 2, \dots, n_c$ ), entonces se puede definir un vector  $\mathbf{a}_{ki}(\mathbf{x}_j)$  dado por (4.69) que es un vector fila de seis componentes asociado al punto  $\mathbf{x}_j$  ( $j = 1, 2, \dots, p_{ki}$ ) de la imagen captada por la cámara  $i$ :

$$\mathbf{a}_k(\mathbf{x}_j) = \left( \frac{S_{i1}}{Z_{ci}}, \frac{S_{i2}}{Z_{ci}}, \frac{S_{i3}}{Z_{ci}}, \mathbf{Q}_{i1} + \frac{\mathbf{R}_{i1}}{Z_{ci}}, \mathbf{Q}_{i2} + \frac{\mathbf{R}_{i2}}{Z_{ci}}, \mathbf{Q}_{i3} + \frac{\mathbf{R}_{i3}}{Z_{ci}} \right) \quad (4.69)$$

Las componentes de velocidad lineal  $\mathbf{v}_{wk}$  y angular  $\boldsymbol{\omega}_{wk}$  pueden obtenerse resolviendo un sistema de ecuaciones lineales como el mostrado en la ecuación (4.70):

$$\mathbf{A}_k \boldsymbol{\rho}_{wk} = \mathbf{b}_k \quad k = 1, \dots, N \quad (4.70)$$

donde  $\boldsymbol{\rho}_{wk} = (\mathbf{v}_{wk}, \boldsymbol{\omega}_{wk})^T$  es el vector que representa las seis componentes del movimiento rígido correspondientes a la región  $\Omega_k$  en el sistema de referencia global ( $\Gamma_w$ ). Considerando la región  $\{\Omega_{ki}\}_{k=1}^{N-1}$  como el conjunto de todos los píxeles contenidos dentro de la curva  $\{\gamma_{ki}\}_{k=1}^{N-1}$  en cada una de las  $n_c$  ( $i=1, 2, \dots, n_c$ ) cámaras disponibles y como región  $\Omega_{Ni}$  a los píxeles correspondientes al fondo de la escena, se pueden definir dos vectores,  $\mathbf{A}_k$  y  $\mathbf{b}_k$ , como sigue:

$$\mathbf{A}_k = \begin{pmatrix} \mathbf{a}_{k1}(x_{11}) \\ \vdots \\ \mathbf{a}_{k1}(x_{p_{k1}1}) \\ \mathbf{a}_{k2}(x_{12}) \\ \vdots \\ \mathbf{a}_{kn_c}(x_{p_{kn_c}n_c}) \end{pmatrix} \quad y \quad \mathbf{c}_k = \begin{pmatrix} -I_{t1}(x_{11}) \\ \vdots \\ -I_{t1}(x_{p_{k1}1}) \\ -I_{t2}(x_{12}) \\ \vdots \\ -I_{tn_c}(x_{p_{kn_c}n_c}) \end{pmatrix} \quad k = 1, \dots, N$$

En caso de que el movimiento de los robots esté restringido al plano del suelo, la ecuación (4.70) puede simplificarse teniendo en cuenta que, como el sistema de referencia global  $\Gamma_w$  está definido sobre el plano del suelo, la componente de velocidad

lineal en  $Z$  debe ser nula (los puntos del robot no cambian su altura). Además, para el caso de la velocidad angular, únicamente la componente  $Z$  será distinta de cero. De esta forma tanto el vector de velocidades  $\mathbf{p}_{wk}$  como el vector  $\mathbf{a}_{ki}(\mathbf{x}_j)$  asociado a cada punto están formados por tres elementos, quedando reducidos a las expresiones mostradas en las ecuaciones (4.71) y (4.72).

$$\mathbf{p}_{wk} = (\mathbf{v}_{wk}^x, \mathbf{v}_{wk}^y, \boldsymbol{\omega}_{wk}^z)^T \quad (4.71)$$

$$\mathbf{a}_k(\mathbf{x}_j) = \left( \frac{S_{i1}}{Z_{ci}}, \frac{S_{i2}}{Z_{ci}}, \mathbf{Q}_{i3} + \frac{\mathbf{R}_{i3}}{Z_{ci}} \right) \quad (4.72)$$

### Estimación de la profundidad mediante descenso por el gradiente

En la segunda etapa la función a minimizar para recuperar la profundidad es la mostrada en la ecuación (4.73), en la que  $\chi_{ki}$  es la función característica de la región  $k$  vista en la imagen de la cámara  $i$  ( $\Omega_{ki}$ ).

$$E(Z) = \sum_{k=1}^N \sum_{i=1}^{n_c} \int_{\Omega_{ki}} [\psi_{ki}^2(\mathbf{x}) + \mu g(\|\nabla Z_{ci}\|)] d\mathbf{x} = \int_{\Omega} \sum_{k=1}^N \sum_{i=1}^{n_c} [\chi_{ki}(\mathbf{x})(\psi_{ki}^2(\mathbf{x}) + \mu g(\|\nabla Z_{ci}\|))] d\mathbf{x} \quad (4.73)$$

La derivada de la función objetivo de la ecuación (4.73) respecto a la profundidad en cada una de las cámaras ( $Z_{ci}$ ,  $i = 1, 2, \dots, n_c$ ) se obtiene siguiendo el mismo proceso que se explicó en el apartado 4.1.3 para el caso de una sola cámara, siendo el resultado el presentado en la ecuación (4.74):

$$\frac{\partial E}{\partial Z_{ci}} = \sum_{k=1}^N \chi_{ki} \left[ \frac{-2}{Z_{ci}^2} (\mathbf{s}_i \mathbf{R}_{wci} \mathbf{v}_{wk} + \mathbf{t}_{wci}^T \mathbf{r}_i \text{adj}(\mathbf{R}_{wci}) \boldsymbol{\omega}_{wk}) \psi_{ki} - \mu \text{div} \left( \frac{\mathbf{g}'(\|\nabla Z_{ci}\|)}{\|\nabla Z_{ci}\|} \nabla Z_{ci} \right) \right] \quad (4.74)$$

Sustituyendo, de nuevo,  $\mathbf{S}_i = \mathbf{s}_i \cdot \mathbf{R}_{wci}$ ,  $\mathbf{Q}_i = \mathbf{q}_i \cdot \text{adj}(\mathbf{R}_{wci})$  y  $\mathbf{R}_i = \mathbf{t}_{wci}^T \cdot \mathbf{r}_i \cdot \text{adj}(\mathbf{R}_{wci})$ , la ecuación anterior puede reescribirse como:

$$\frac{\partial E}{\partial Z_{ci}} = \sum_{k=1}^N \chi_{ki} \left[ \frac{-2}{Z_{ci}^2} (\mathbf{S}_i \mathbf{v}_{wk} + \mathbf{R}_i \boldsymbol{\omega}_{wk}) \psi_{ki} - \mu \text{div} \left( \frac{\mathbf{g}'(\|\nabla Z_{ci}\|)}{\|\nabla Z_{ci}\|} \nabla Z_{ci} \right) \right] \quad (4.75)$$

Dado un conjunto de curvas  $\{\gamma_{ki}\}_{k=1}^{N-1}$  por cada una de las  $n_c$  cámaras, que dividen la imagen correspondiente en  $N$  regiones  $\{\Omega_{ki}\}_{k=1}^N$ , se obtiene la expresión (4.76) para las derivadas de la función objetivo respecto de la profundidad para cada región de cada una de las cámaras disponibles.

$$\frac{\partial E}{\partial Z_{ci}} = \sum_{k=1}^N \chi_{ki} \left[ \frac{-2}{Z_{ci}^2} (\mathbf{S}_i \mathbf{v}_{wk} + \mathbf{R}_i \boldsymbol{\omega}_{wk}) \psi_{ki} - \mu \operatorname{div} \left( \frac{\mathbf{g}'(\|\nabla Z_{ci}\|)}{\|\nabla Z_{ci}\|} \nabla Z_{ci} \right) \right] \quad (4.76)$$

A estas ecuaciones se les añade una condición de contorno de Neumann  $\partial Z/\partial \mathbf{n} = 0$ , siendo  $\mathbf{n}$  el vector unitario normal al borde de la región  $\Omega_k$ . Las ecuaciones de descenso correspondientes a (4.76), para las  $N$  regiones vistas desde cada una de las  $n_c$  cámaras consideradas, son las mostradas en (4.77), donde  $\tau$  indica el tiempo de ejecución del algoritmo y  $\mathbf{g}'$  es la derivada ordinaria de la función de preservación de bordes  $\mathbf{g}$ .

$$\frac{\partial Z_{ci}}{\partial \tau} = \frac{2}{Z_{ci}^2} (\mathbf{S}_i \mathbf{v}_{wk} + \mathbf{R}_i \boldsymbol{\omega}_{wk}) \psi_{ki} + \mu \operatorname{div} \left( \frac{\mathbf{g}'(\|\nabla Z_{ci}\|)}{\|\nabla Z_{ci}\|} \nabla Z_{ci} \right) \quad \begin{array}{l} i = 1, \dots, n_c \\ k = 1, \dots, N \end{array} \quad (4.77)$$

Como ya se comentó al explicar el proceso de minimización de energía para una cámara en el apartado 4.1.3, en la literatura existen diferentes funciones de preservación de bordes ( $\mathbf{g}(\mathbf{a})$ ). Algunas de estas funciones son analizadas en el capítulo 7, en el que se ha determinado que todas ellas son igualmente válidas, aunque requieren un ajuste adecuado de la constante  $\mu$ . Por simplicidad se ha decidido emplear una función cuadrática ( $\mathbf{g}(\mathbf{a}) = \mathbf{a}^2$ ). Con respecto a la elección del valor de las constantes  $\lambda$  y  $\mu$  para el caso de  $n_c$  cámaras, se ha realizado un estudio que se describe en detalle en el apartado 7.3.

### Evolución de las curvas para segmentación de movimiento 3D

Conocida la profundidad  $Z_{ci}$  y los parámetros del movimiento rígido 3D  $\{\mathbf{v}_{wk}, \boldsymbol{\omega}_{wk}\}_{k=1}^N$  (las componentes de velocidad lineal y angular referidas a  $\Gamma_w$ ) para cada una de las  $N$  regiones en las que se ha dividido la imagen, la función objetivo a minimizar, con respecto a las curvas que definen los contornos de los robots móviles presentes en cada imagen  $\{\gamma_{ki}\}_{k=1, \dots, N-1}^{i=1, \dots, n_c}$ , es la mostrada en la ecuación (4.78), donde  $\xi_{ki}(\mathbf{x}) = \psi_{ki}^2(\mathbf{x}) + \mu \mathbf{g}(\|\nabla Z_{ci}\|)$ .

$$E[\{\gamma_{ki}\}_{k=1, \dots, N-1}^{i=1, \dots, n_c}] = \sum_{k=1}^N \sum_{i=1}^{n_c} \int_{\Omega_{ki}} \xi_{ki}(\mathbf{x}) d\mathbf{x} + \lambda \sum_{k=1}^{N-1} \sum_{i=1}^{n_c} \oint_{\gamma_{ki}} ds \quad (4.78)$$

Para la segmentación de múltiples regiones se obtienen las siguientes derivadas para las imágenes captadas por cada una de las  $n_c$  cámaras:

$$\frac{\partial E_{ki}}{\partial \gamma_{ki}} = \left( \xi_{ki} - \varphi_{ki} + \lambda \kappa_{\gamma_{ki}} \right) \mathbf{n}_{ki}, \quad \begin{array}{l} i = 1, \dots, n_c \\ k = 1, \dots, N-1 \end{array} \quad (4.79)$$

y las correspondientes ecuaciones de descenso de *Euler-Lagrange*:

$$\frac{\partial \gamma_{ki}}{\partial \tau}(s, \tau) = - \left( \xi_{ki}(\gamma_{ki}(s)) - \varphi_{ki}(\gamma_{ki}(s)) + \lambda \kappa_{\gamma_{ki}}(\gamma_{ki}(s)) \right) \times \mathbf{n}_{ki}(\gamma_{ki}(s)) \quad (4.80)$$

$$i = 1, \dots, n_c \quad k = 1, \dots, N-1$$

donde  $\kappa_{\gamma_{ki}}$  es la curvatura media del contorno definido por la curva  $\gamma_{ki}$ ,  $\mathbf{n}_{ki}$  es la función unitaria normal al exterior la citada curva y las funciones  $\varphi_{ki}$  se definen como:

$$\varphi_{ki}(\gamma_{ki}(s)) = \min_{j \neq k} \xi_{ji}(\gamma_{ki}(s)) \quad (4.81)$$

Para la implementación se representa cada curva  $\gamma_{ki}$  mediante el conjunto de nivel cero de una función  $\Phi_{ki}$ , de forma que el interior de cada curva  $\gamma_{ki}$  se corresponde con el conjunto de puntos en los que  $\Phi_{ki}$  es mayor que cero ( $\{\Phi_{ki} > 0\}$ ) mientras que el exterior de la citada curva se corresponde con los puntos para los que la función  $\Phi_{ki}$  es inferior a cero. De esta forma se obtiene el siguiente sistema de ecuaciones diferenciales parciales acopladas:

$$\frac{\partial \Phi_{ki}}{\partial \tau}(\mathbf{x}, \tau) = - \left( \xi_{ki}(\mathbf{x}) - \varphi_{ki}(\mathbf{x}) + \lambda \kappa_{\Phi_{ki}}(\mathbf{x}) \right) \|\nabla \Phi_{ki}(\mathbf{x})\| \quad (4.82)$$

$$i = 1, \dots, n_c \quad k = 1, \dots, N-1$$

donde la curvatura media  $\kappa_{\Phi_{ki}}$  se define como:  $div(\nabla \Phi_{ki} / \|\nabla \Phi_{ki}\|)$  y la función  $\varphi_{ki}(x)$  es la mostrada en la ecuación (4.83):

$$\varphi_{ki}(\mathbf{x}) = \begin{cases} \min_{j \neq k} \xi_j(\mathbf{x}) & \text{for } j \neq k \in \{1, \dots, N-1\}, \Phi_{ji}(\mathbf{x}) > 0 \\ \xi_{Ni}(\mathbf{x}) & \text{else} \end{cases} \quad (4.83)$$

Las tres etapas descritas en este apartado se repiten hasta alcanzar la convergencia del algoritmo. Se considera que el algoritmo converge cuando las variables implicadas dejan de evolucionar de forma significativa.

Tanto el número de iteraciones necesarias hasta la convergencia del algoritmo de minimización de energía, como la calidad final de los resultados obtenidos (segmentación y parámetros de movimiento 3D) dependen fuertemente tanto de la inicialización de las curvas como de las constantes de ponderación de cada uno de los

términos de la función objetivo (ecuación (4.66))  $\lambda$  y  $\mu$  y la función de preservación de bordes ( $g(s)$ ) elegida.

Para determinar el efecto de las constantes  $\lambda$  y  $\mu$  así como de la función de preservación de bordes se han realizado un conjunto de pruebas experimentales que se presentan a continuación. Respecto al efecto de la inicialización de las curvas, este ha sido estudiado en detalle a lo largo del capítulo 7 de este documento.

#### 4.2.3.2. Criterio de convergencia

Como ya se ha comentado anteriormente, se considera que el algoritmo de minimización de energía converge cuando las variables implicadas en el mismo dejan de evolucionar de forma significativa entre una iteración y la siguiente. Esto nos permite establecer una condición de parada del algoritmo.

Dado que la función objetivo propuesta para  $n_c$  cámaras (mostrada en la ecuación (4.66)) depende de tres grupos de parámetros: las curvas que definen los contornos de los  $(N-1)$  robots móviles presentes en las escenas captadas por cada una de las  $n_c$  cámara:  $\{\gamma_{ki}\}_{k=1, \dots, N-1}^{i=1, \dots, n_c}$ , las componentes de la velocidad lineal y angular de cada uno de los  $(N-1)$  robots móviles, así como del fondo de la escena en el sistema de referencia del mundo  $\Gamma_w$ :  $\{\mathbf{v}_{wk}\}_{k=1}^N$ ,  $\{\boldsymbol{\omega}_{wk}\}_{k=1}^N$  (comunes para todas las cámaras) y la distancia de cada punto 3D ( $\mathbf{P}$ ) a cada una de las cámaras ( $i=1, \dots, n_c$ ), es necesario estudiar la evolución de cada uno de estos grupos de forma independiente. Se considera que el algoritmo ha alcanzado la convergencia, únicamente si la variación de todos los grupos de variables es suficientemente pequeña.

Teniendo en cuenta que los grupos de variables implicados en la función objetivo son los mismos que en el caso de una cámara, el criterio de convergencia utilizado es el mismo que se ha definido en el apartado 4.1.3.2. Si bien en este caso hay que hacer la salvedad de que las variaciones en la profundidad y las curvas se obtienen como el valor medio de la variación de los citados parámetros para las imágenes captadas para las  $n_c$  cámaras consideradas, y las componentes de velocidad están expresadas en el sistema de referencia global ( $\Gamma_w$ ).

Los valores que determinan la convergencia de cada uno de los grupos de variables implicados en la función objetivo para  $n_c$  cámaras, (4.66), se resumen en la Tabla 4.2.

El hecho de utilizar el mismo criterio de convergencia para este caso que para el de una única cámara permite comparar los resultados obtenidos en ambos casos. Tal como se podrá comprobar en el capítulo de resultados.

Grupo de Variables	Criterio de convergencia
Velocidad lineal: $\{\mathbf{v}_{wki}\}_{k=1,\dots,N}^{i=1,\dots,n_c}$	Variación inferior a 0.5 mm.
Velocidad angular: $\{\boldsymbol{\omega}_{wki}\}_{k=1,\dots,N}^{i=1,\dots,n_c}$	Variación inferior a 0.5 °
Profundidad: $Z_{ci} \ i=1,\dots,n_c$	Variación inferior a 10 mm.
Curvas: $\{\gamma_{ki}\}_{k=1,\dots,N-1}^{i=1,\dots,n_c}$	Variación inferior a 3 píxeles

Tabla 4.2. Criterio de convergencia: valor máximo de variación de la variable entre una iteración y la siguiente para el que se considera que la variable converge para cada uno de los grupos de variables implicados en la función objetivo de la ecuación (4.66).

### 4.3. Conclusiones y aportaciones

En este capítulo se han presentado dos alternativas para la segmentación de movimiento y el posicionamiento 3D de múltiples robots móviles en un espacio inteligente. Esto se realiza a partir de las imágenes adquiridas por un conjunto de cámaras fijas y calibradas. Tanto la segmentación como la posición 3D se obtienen, en ambos casos, mediante la minimización de una función objetivo.

En la primera alternativa, se obtiene la segmentación y posición 3D, relativa a cada una de las cámaras, de forma independiente. Posteriormente, se incluye un proceso de fusión que permite incorporar la información de todas las cámaras. De esta forma se obtiene un resultado global. Sin embargo, en la minimización de la función objetivo, que se realiza para cada cámara, únicamente se tiene en cuenta la información de una de ellas. Por este motivo, los resultados obtenidos empleando esta alternativa son poco robustos.

Con objeto de incrementar la robustez y mejorar la precisión de los resultados de la segmentación y posicionamiento 3D se ha propuesto una nueva función objetivo que incorpora la información adquirida por múltiples cámaras. La minimización de esta función permite obtener la segmentación de movimiento y estimar la posición 3D de múltiples robots móviles teniendo en cuenta la información obtenida de forma simultánea de todas las cámaras disponibles.

La propuesta de la función objetivo para el caso de múltiples cámaras, así como el algoritmo de minimización de la misma, son las aportaciones más notables realizadas en este capítulo.

Otras aportaciones destacables son, por una parte, la propuesta de una solución que permite realizar la fusión de los resultados en caso de que la minimización de la función objetivo se ejecute de forma independiente por cada una de las cámaras, y, por otra parte, se ha definido un criterio de convergencia del proceso de minimización para la aplicación de interés de esta tesis. Además, este criterio es el mismo para las dos soluciones propuestas, permitiendo comparar los resultados obtenidos en ambos casos.



## Capítulo 5

# **Inicialización de las curvas que definen los contornos de los robots móviles**



El objetivo de la etapa de inicialización es la obtención de un conjunto de curvas sobre las imágenes captadas, cada una de las cuales defina el contorno de uno de los robots móviles presentes en la escena.

El primer paso para la obtención de las curvas es la detección de los diferentes robots móviles. En nuestro caso, debido a que tenemos un conjunto de cámaras fijas ubicadas en el entorno, se propone modelar el fondo de la escena (proceso *off-line*). A partir de este modelo y de las imágenes captadas *on-line* se determina qué objetos nuevos han aparecido en la escena después de la obtención del modelo de fondo.

Tanto para el modelado del fondo, como para la comparación de las imágenes de entrada con el citado modelo se emplea una generalización del análisis de componentes principales (PCA). Esta técnica permite representar un conjunto de  $k$  observaciones de  $n$  variables con un número menor de variables,  $m$ , construidas como combinaciones lineales de las originales [Turk & Pentland 1991]. El análisis de componentes principales ha demostrado su eficacia en numerosos entornos: [Martínez & Kak 2001], [Maciejewski & Roberts 2001], [Chen et al. 2002], algunos de ellos relacionados con procesamiento de imágenes.

En este trabajo, dada la estructura matricial de las imágenes, se ha decidido emplear la técnica *GPCA* (*Generalized Principal Components Analysis*) descrita en [Jieping et al 2004] en lugar de PCA. La ventaja de GPCA [Jieping et al 2004] sobre el PCA clásico [Turk & Pentland 1991] radica en que el primero emplea una representación matricial (en lugar de vectorial) de los datos, de forma que conserva la información de vecindad de los píxeles en las imágenes. Esto permite conseguir una reducción en los requerimientos de memoria así como una disminución en el tiempo de ejecución del algoritmo. Además, en el caso de aplicar estas técnicas para la compresión de imágenes, empleando el mismo número de componentes principales, la calidad de la imagen comprimida usando GPCA es mejor que en el caso de PCA.

En la Figura 5.1 se presenta el diagrama general del proceso de inicialización de las curvas que, como se puede observar, consta de tres etapas bien diferenciadas:

1. En la primera de estas etapas, se realiza el modelado del fondo mediante GPCA, con la escena sin robots. Esta etapa puede realizarse *off-line* debido a que las cámaras se encuentran ubicadas en posiciones fijas del entorno.

2. A continuación, en una segunda etapa que se realiza *on-line*, cada una de las imágenes de entrada pertenecientes a la secuencia que se desea segmentar se compara con el modelo de fondo obtenido anteriormente empleando, de nuevo, la técnica GPCA. La diferencia entre la imagen de entrada y el modelo de fondo se umbraliza para determinar qué píxeles son candidatos a pertenecer a robots móviles (nuevos objetos en la escena), y cuáles corresponden al fondo. Es muy importante la correcta elección del valor del umbral a aplicar.
3. Finalmente, con objeto de minimizar el efecto de las sombras que proyectan los diferentes objetos, se añade una última etapa para la eliminación de las mismas. Dado que se tienen varias imágenes de la escena, en caso de que los robots se muevan sobre un plano, es posible eliminar las sombras de forma sencilla por la altura de las mismas. Sin embargo, se ha decidido contemplar la posibilidad de que la superficie por la que se desplazan los robots no sea plana. En estas condiciones, la eliminación de sombras se realiza mediante la proyección de cada imagen de entrada a un espacio invariante a la iluminación, según se explicará más adelante.

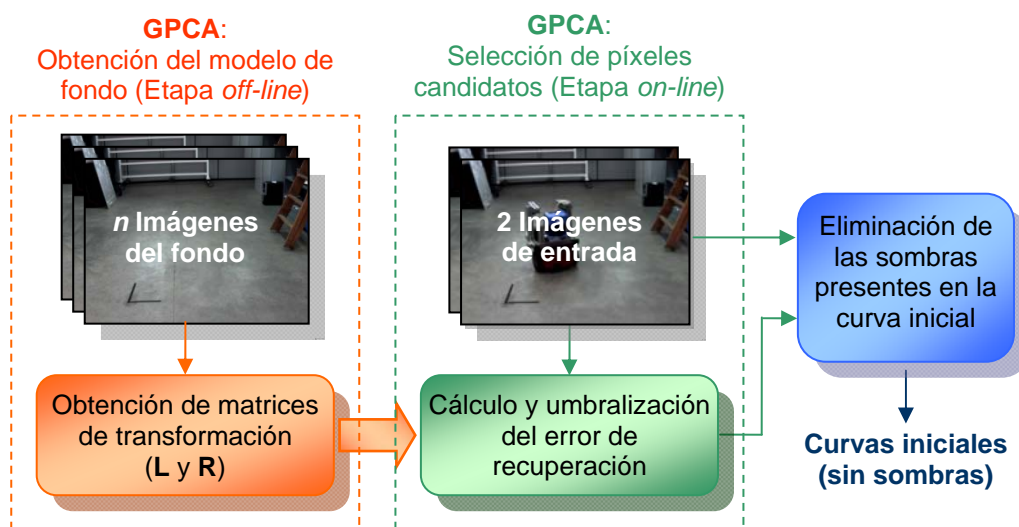


Figura 5.1. Diagrama de bloques general en el que se presentan las diferentes etapas necesarias para la inicialización de las curvas que definen los contornos de los robots móviles presentes en la escena.

Las diferentes etapas del proceso de inicialización de las curvas se describen en los siguientes apartados de este capítulo.

## 5.1. Obtención del modelo de fondo de la escena

En esta primera etapa del algoritmo, la técnica GPCA [Jieping et al 2004] permite obtener dos matrices de transformación  $\mathbf{L}$  y  $\mathbf{R}$ , y la matriz media  $\mathbf{M}$  a partir de un conjunto de  $N_i$  imágenes del fondo de la escena  $\{\mathbf{I}_j\}_{j=1}^{N_i}$  en las que no aparece ningún objeto en movimiento. Mediante las matrices obtenidas, es posible representar la imagen del fondo en un espacio transformado GPCA conservando la información más significativa contenida en la citada imagen. Posteriormente se puede recuperar dicha imagen al espacio original. La calidad de las imágenes en el espacio transformado depende fuertemente del número de componentes principales ( $d$ ) que se tienen en cuenta para la transformación y que coincide con la dimensión del espacio transformado.

El primer paso para la obtención del modelo de fondo es estimar la media  $\mathbf{M}$  de las  $N_i$  imágenes, (5.1), y a continuación restar a cada una de las imágenes la matriz media calculada, (5.2).

$$\mathbf{M} = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{I}_j \quad (5.1)$$

$$\tilde{\mathbf{I}}_j = \mathbf{I}_j - \mathbf{M} \quad j = 1..N_i \quad (5.2)$$

Estas imágenes a las que se les ha sustraído la media son las que se utilizan en las diferentes etapas del algoritmo GPCA para la obtención de las matrices de transformación  $\mathbf{L}$  y  $\mathbf{R}$ . Dicho algoritmo consiste en un proceso iterativo que tras la inicialización de la matriz  $\mathbf{L}_0$  se repite hasta alcanzar la convergencia.

En cada una de las iteraciones, se comienza calculando dos matrices, una  $\mathbf{M}_R$  obtenida a partir de las imágenes a las que se ha restado la media, y la otra  $\mathbf{L}_{i-1}$  obtenida en la iteración anterior. Posteriormente se forma la matriz  $\mathbf{R}_i$  cuyas columnas son los autovectores  $\{\phi_j^R\}_{j=1}^d$  asociados a los  $d$  mayores autovalores de la matriz  $\mathbf{M}_R$  según (5.3). Una vez obtenida la matriz  $\mathbf{R}_i$ , las columnas de  $\mathbf{L}_i$  se determinan como los autovectores  $\{\phi_j^L\}_{j=1}^d$  asociados a los  $d$  mayores autovalores de una matriz  $\mathbf{M}_L$  obtenida según (5.4).

$$\mathbf{M}_R = \sum_{j=1}^N \tilde{\mathbf{I}}_j^T \mathbf{L}_i \mathbf{L}_i^T \tilde{\mathbf{I}}_j \Rightarrow \mathbf{R}_i = [\phi_1^R, \dots, \phi_d^R] \quad (5.3)$$

$$\mathbf{M}_L = \sum_{j=1}^N \tilde{\mathbf{I}}_j^T \mathbf{R}_i \mathbf{R}_i^T \tilde{\mathbf{I}}_j \Rightarrow \mathbf{L}_i = [\phi_1^L, \dots, \phi_d^L] \quad (5.4)$$

La convergencia de este algoritmo viene determinada por el cambio en el error cuadrático medio residual (*RMSE*) definido en cada iteración según la ecuación (5.5). Se considera que el algoritmo converge cuando la diferencia entre el *RMSE* en la iteración actual y en la anterior es inferior a un umbral  $\eta > 0$ , establecido previamente de acuerdo con la ecuación (5.6). La convergencia del algoritmo GPCA está garantizada ya que, según se demuestra en [Jieping et al. 2004], el valor del *RMSE* tiene un comportamiento monótonamente decreciente en cada iteración. En esta situación, las matrices de transformación utilizadas en GPCA (**L** y **R**) serán las matrices  $\mathbf{L}_i$  y  $\mathbf{R}_i$  obtenidas en la iteración en la que el algoritmo alcanza la convergencia.

$$RMSE(i) = \sqrt{\frac{1}{n} \sum_{j=1}^N \|\tilde{\mathbf{I}}_j - \mathbf{L}_i \mathbf{L}_i^T \tilde{\mathbf{I}}_j \mathbf{R}_i \mathbf{R}_i^T\|_F^2} \quad (5.5)$$

$$RMSE(i) - RMSE(i-1) \leq \eta \quad (5.6)$$

En la Figura 5.2 se muestran, en forma de diagrama de bloques, todas las etapas necesarias para la obtención del modelo de fondo usando el algoritmo GPCA.

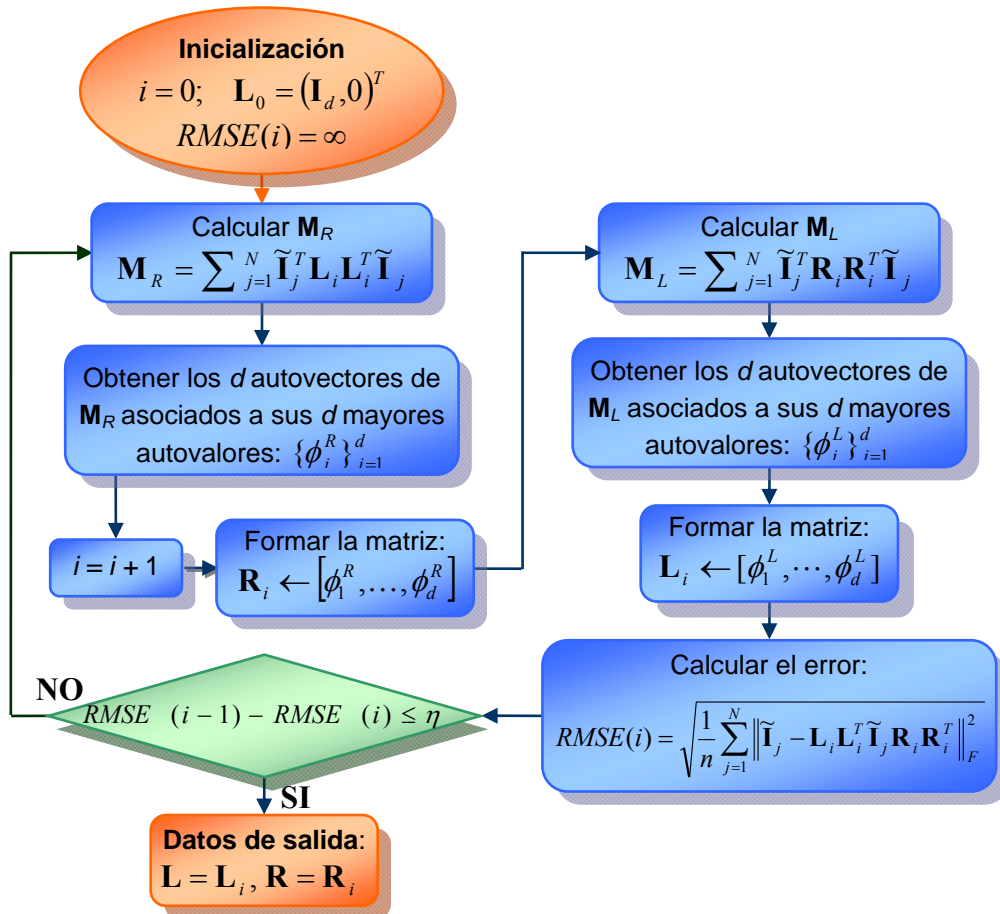


Figura 5.2. Obtención del modelo de fondo de la escena usando GPCA a partir de un conjunto de imágenes de fondo en las que no aparecen robots móviles. Esta etapa sólo se ejecuta una vez y puede realizarse *off-line*

Tanto el número de imágenes consideradas para la creación del modelo de fondo, como el número de componentes principales ( $d$ ) determinan la calidad de la imagen transformada. El número de imágenes de fondo debe ser suficientemente alto para contemplar los posibles cambios que se den en la escena, especialmente los debidos a la iluminación. En la situación contemplada en este trabajo, en el que las imágenes se capturan en un entorno interior controlado, la iluminación no varía de forma significativa en las secuencias de imágenes. Por este motivo, cabe esperar que aplicaciones reales con un número pequeño de imágenes sea posible modelar adecuadamente el fondo de la escena. Con respecto al número de componentes principales que se consideran ( $d$ ), su valor determina la dimensión del espacio transformado y, consecuentemente, la cantidad de información que se conserva tras la transformación.

A continuación se presentan las pruebas realizadas para determinar el efecto del número de imágenes y autovectores utilizados para la obtención del modelo de fondo.

### **5.1.1. Elección del número de imágenes y de autovectores utilizados en la obtención del modelo de fondo usando GPCA**

Como ya se ha comentado, tanto el número de imágenes, como el número de autovectores influyen en la calidad del modelo de fondo calculado usando GPCA.

Para determinar el efecto de estos parámetros se ha realizado un conjunto de pruebas experimentales en las que se ha medido el valor medio del error de recuperación de 15 imágenes de fondo. Este error se define como la diferencia entre la imagen de entrada y la imagen recuperada. Para la transformación y posterior recuperación de la imagen se han utilizado las matrices de transformación GPCA ( $\mathbf{L}$  y  $\mathbf{R}$ ) y la matriz media  $\mathbf{M}$  de las  $N_i$  imágenes de fondo obtenidas utilizando diferentes valores del número de imágenes y de autovectores.

En la Tabla 5.1 se presentan los valores medios de los errores de recuperación obtenidos para los diferentes valores de los parámetros bajo estudio. En todos los casos, el error de recuperación medio se ha obtenido como el valor medio de la diferencia

entre la imagen de entrada y la imagen recuperada al cuadrado. El valor presentado en la tabla es el valor medio del error de recuperación calculado para 15 imágenes del fondo.

	<b>1 Imag.</b>	<b>2 Imág.</b>	<b>5 Imág.</b>	<b>10 Imág.</b>	<b>15 Imág.</b>	<b>20 Imág.</b>
1 autovect.	2.466	1.8225	1.4297	1.2973	1.2513	1.2738
2 autovect.	-	1.8215	1.4286	1.2961	1.2498	1.2724
3 autovect.	-	-	1.4277	1.2944	1.2484	1.2711
4 autovect.	-	-	1.4262	1.2934	1.2471	1.2696
5 autovect.	-	-	1.4252	1.2922	1.2437	1.2674
6 autovect.	-	-	-	1.291	1.2421	1.2641
7 autovect.	-	-	-	1.2889	1.2407	1.2627
8 autovect.	-	-	-	1.2872	1.2393	1.2612
9 autovect.	-	-	-	1.2861	1.2378	1.2595
10 autovect.	-	-	-	1.2838	1.2359	1.2583
11 autovect.	-	-	-	-	1.2343	1.2571
12 autovect.	-	-	-	-	1.2328	1.2558
13 autovect.	-	-	-	-	1.2308	1.2541
14 autovect.	-	-	-	-	1.2289	1.2524
15 autovect.	-	-	-	-	1.2271	1.2508
16 autovect.	-	-	-	-	-	1.2491
17 autovect.	-	-	-	-	-	1.2471
18 autovect.	-	-	-	-	-	1.2453
19 autovect.	-	-	-	-	-	1.2432
20 autovect.	-	-	-	-	-	1.2415

Tabla 5.1. Valores de los errores de recuperación en función del número de imágenes y autovectores considerados en el cálculo del modelo de fondo usando GPCA

En la Figura 5.3 se puede observar gráficamente el valor del error de recuperación en función del número de imágenes utilizadas en el cálculo del modelo de fondo para 1, 2 y 5 autovectores. Se puede observar como el valor del error desciende a medida que aumenta el número de imágenes. Este descenso es especialmente significativo entre 1 y 10 imágenes, tendiendo a mantenerse casi constante a partir de este valor. Con respecto al efecto del número de autovectores, en la figura también se puede observar como el hecho de aumentar el número de autovectores de 1 a 5 no influye de forma significativa en el valor del error de recuperación de las imágenes de fondo.



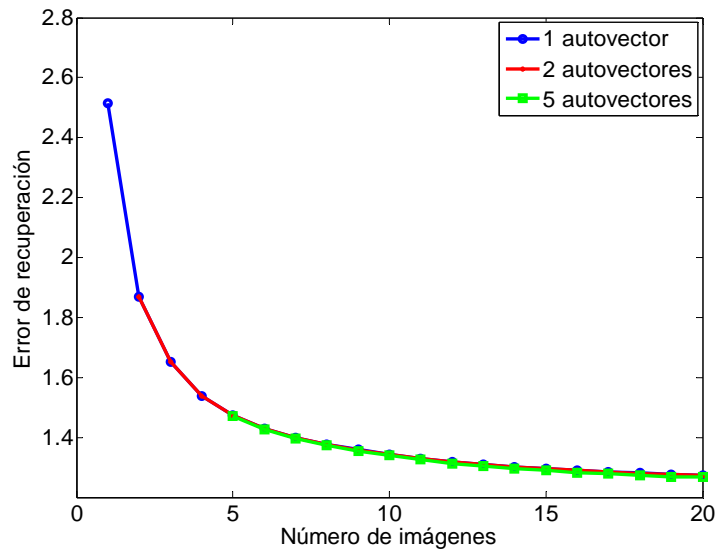


Figura 5.3. Error de recuperación de las imágenes de fondo en función del número de imágenes utilizadas para el cálculo del modelo de fondo y utilizando diferente número de autovectores.

Tanto el número de imágenes como el de autovalores considerados influyen en el tiempo de cómputo consumido en el cálculo de las matrices de transformación  $L$  y  $R$ . En la Figura 5.4 se ha representado este tiempo en función del número de imágenes para 1, 2 y 5 autovectores. Se puede observar como a medida que aumenta el número de imágenes, lo hace también el tiempo de cómputo. En el caso de un único autovalor (línea en azul en la figura) se puede observar como el incremento del tiempo con el número de imágenes es aproximadamente lineal, sin embargo, para 2 y 5 autovectores aparecen algunos picos. Esto se debe a que el algoritmo GPCA utilizado en el modelado del fondo es iterativo (Figura 5.2) de forma que el tiempo consumido varía no solo en función del número de imágenes y autovectores, sino también en función del número de iteraciones necesarias para la convergencia.

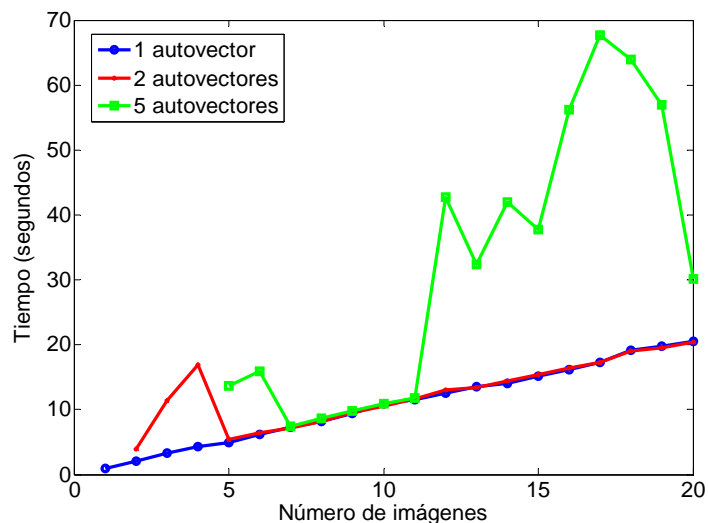


Figura 5.4. Tiempo consumido en la obtención del modelo de fondo en función del número de imágenes utilizadas para el cálculo, para 1, 2 y 5 autovectores.

A la vista de los resultados anteriores, se ha decidido fijar en 10 el número de imágenes utilizadas para el cálculo del modelo de fondo como solución de compromiso entre el valor del error de recuperación y el tiempo de cómputo consumido.

Con respecto al número de autovectores, dado que el error apenas disminuye a medida que aumentamos el número de autovectores, mientras que el tiempo aumenta considerablemente, se ha decidido emplear únicamente el autovector asociado al mayor autovalor. Es decir, el espacio transformado GPCA utilizado será dimensiones  $1 \times 1$ .

Para evaluar la viabilidad del uso de 10 imágenes y un único autovector en la generación del modelo de fondo usando GPCA, en la Figura 5.5(a) y (b) se muestran dos imágenes de fondo en escala de grises, mientras en la Figura 5.5(c) y (d) se presenta la imagen recuperada GPCA obtenida tras proyectar la imagen de entrada al espacio transformado GPCA usando las matrices de transformación ( $\mathbf{L}$ ,  $\mathbf{R}$ ) (5.7) y recuperar la imagen en el espacio original mediante la transformación inversa (5.8). Finalmente, en la Figura 5.5(e) y (f) se ha representado el error de recuperación en cada píxel de la imagen, calculado como la diferencia entre la imagen original y la recuperada al cuadrado.

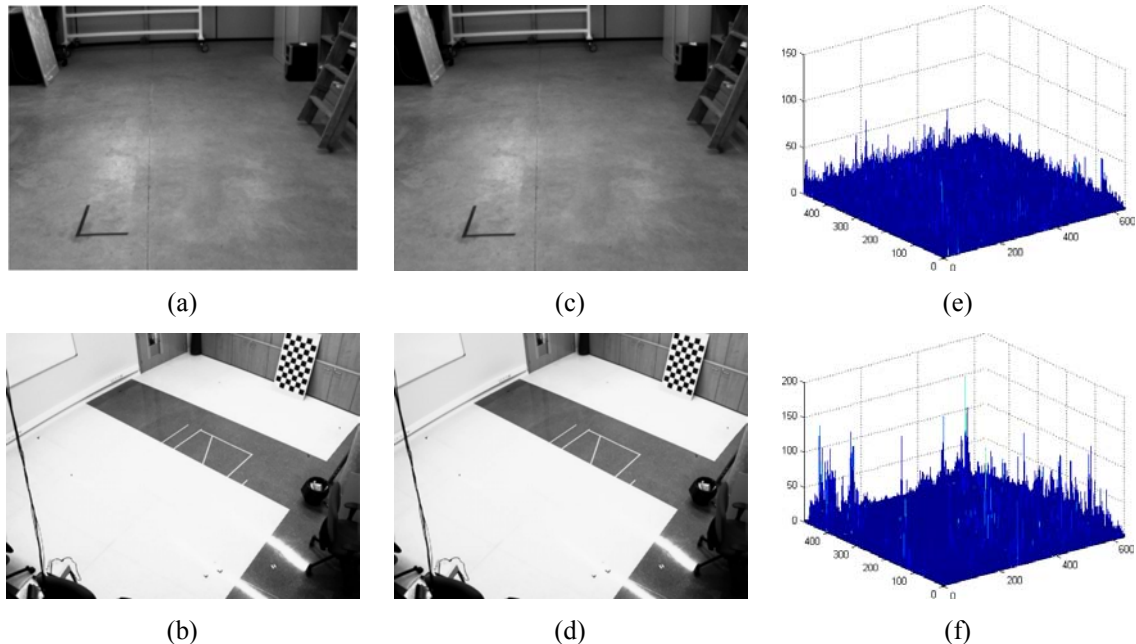


Figura 5.5. (a) y (b) Imágenes del fondo de la escena. (c) y (d) Resultado de la proyección y posterior recuperación de las imágenes (a) y (b) mediante las matrices de transformación GPCA obtenidas a partir de 10 imágenes del fondo con 1 único autovector. (e) y (f) Error de recuperación obtenido como la diferencia entre la imagen original y la recuperada elevada al cuadrado.

### 5.1.2. Comparación entre las técnicas PCA y GPCA para la obtención del modelo de fondo

Como se ha comentado al principio de este capítulo, en esta tesis se ha decidido utilizar la técnica GPCA [Jieping et al. 2004] en lugar de PCA [Turk & Pentland 1991] debido a que GPCA maneja los datos de entrada en formato matricial, conservando la información de vecindad. Esto se traduce en una reducción de los requerimientos de memoria (las matrices con que se trabaja son de dimensiones mucho menores) así como en la mejora de la precisión.

A continuación se presentan los resultados de algunos de los experimentos que se han realizado para evaluar la mejora de GPCA frente a PCA. Indicar que para las pruebas realizadas ha sido necesario reducir el tamaño de las imágenes de entrada mediante el diezmado de las mismas, para evitar los problemas de falta de memoria debido al gran tamaño de las matrices manejadas por la técnica PCA. En todas las pruebas realizadas, las imágenes de entrada han sido las mismas para ambas técnicas.

#### Tamaño de las matrices

En primer lugar se ha analizado el tamaño de las matrices que se almacenan en memoria durante el procesamiento de las imágenes tanto en el caso de PCA como en el de GPCA. En el caso de PCA, es necesario almacenar en memoria la matriz de covarianza de las imágenes de entrada cuyas dimensiones son de  $(f \times c) \times (f \times c)$  así como la matriz de transformación cuyas dimensiones  $(f \times c) \times d$ . Mientras que la técnica GPCA requiere almacenar cuatro matrices durante el proceso de obtención del modelo de fondo:  $\mathbf{L}$ ,  $\mathbf{R}$ ,  $\mathbf{M}_L$  y  $\mathbf{M}_R$  cuyas dimensiones son:  $f \times d$ ,  $c \times d$ ,  $f \times f$  y  $c \times c$  respectivamente. Todas ellas mucho menores que la matriz de covarianza de PCA. En la Tabla 5.2 se presenta el número de elementos de las diferentes matrices de PCA y GPCA para imágenes de entrada de  $480 \times 640$  píxeles, suponiendo diferentes valores del número de autovectores. En la Tabla 5.2, para el caso de PCA se han tenido en cuenta la matriz de covarianza ( $\mathbf{C}$ ) y la de transformación ( $\mathbf{U}$ ), mientras que para GPCA se han considerado las cuatro matrices involucradas en el algoritmo iterativo para la obtención de las matrices de transformación.

		d=1	d=2	d=5	d=10
PCA	C	$(480 \times 640) \times (480 \times 640) = 307200 \times 307200 = 9.4372 \cdot 10^{10}$			
	U	$(480 \times 640) \times 1 = 307200$	$(480 \times 640) \times 2 = 614400$	$(480 \times 640) \times 5 = 1536000$	$(480 \times 640) \times 10 = 3072000$
	<b>Total</b>	<b><math>9.4372 \cdot 10^{10}</math></b>	<b><math>9.4372 \cdot 10^{10}</math></b>	<b><math>9.4373 \cdot 10^{10}</math></b>	<b><math>9.4375 \cdot 10^{10}</math></b>
GPCA	L	$480 \times 1 = 480$	$480 \times 2 = 960$	$480 \times 5 = 2400$	$480 \times 10 = 4800$
	R	$640 \times 1 = 640$	$640 \times 2 = 1280$	$640 \times 5 = 3200$	$640 \times 10 = 6400$
	$M_L$	$480 \times 480 = 203400$			
	$M_R$	$640 \times 640 = 409600$			
	<b>Total</b>	<b>641120</b>	<b>642240</b>	<b>645600</b>	<b>651200</b>

Tabla 5.2. Número de elementos de las matrices implicadas en la obtención del modelo de fondo utilizando las técnicas PCA y GPCA.

### Tiempo de cómputo

Por otro lado, a pesar de que las matrices de transformación de GPCA se obtienen utilizando un algoritmo iterativo, el tiempo de cómputo de este algoritmo es siempre menor que en el caso de PCA ya que, según se demuestra en [Jieping et al. 2004], el valor del RMSE es monótonamente decreciente con cada iteración, y el algoritmo converge en pocas iteraciones. En la Figura 5.6 se muestra el tiempo de cómputo consumido para la obtención de las matrices de transformación tanto para el caso de PCA (en negro) como el de GPCA (en rojo) en función del número de imágenes y autovectores utilizados en la generación del modelo de fondo.

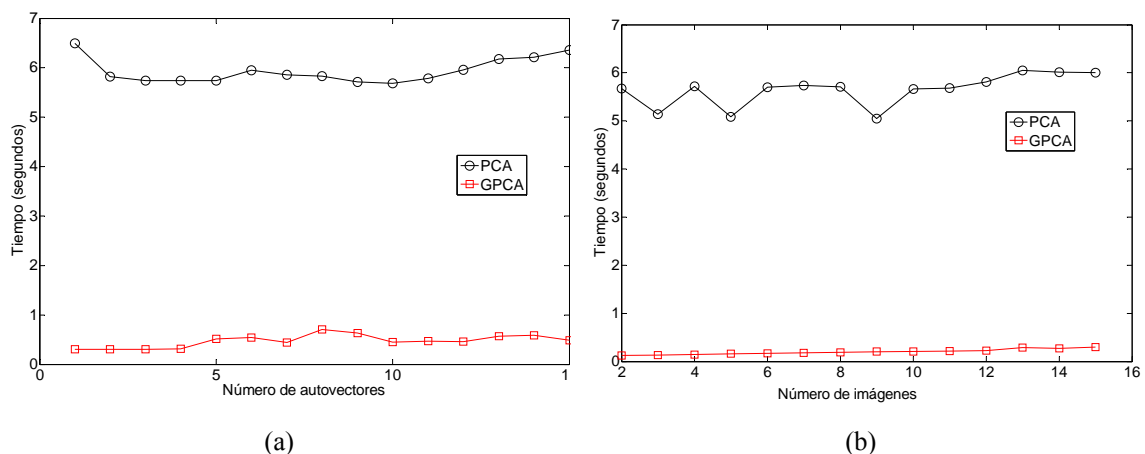


Figura 5.6. Tiempo de cómputo para el cálculo de las matrices de transformación PCA y GPCA (a) en función del número de autovectores y (b) en función del número de imágenes utilizadas.

En la Tabla 5.3 se recogen, a modo de resumen, los valores de tiempo obtenidos para diferentes valores del número de imágenes y autovectores. Tanto en la Figura 5.6 como en la Tabla 5.3 se puede observar que el tiempo de cómputo consumido para la obtención de las matrices de transformación es muy inferior (del orden de 10 veces menor en el peor de los casos) con la técnica GPCA frente a PCA independientemente del número de imágenes y autovectores considerados.

	2 Imágenes		5 Imágenes		10 Imágenes		15 Imágenes	
	PCA	GPCA	PCA	GPCA	PCA	GPCA	PCA	GPCA
<b>1 Autovect.</b>	5.5113	0.1219	5.0397	0.1544	5.7446	0.2101	5.864	0.2724
<b>2 Autovect.</b>	5.0659	0.1256	5.0565	0.1538	5.6888	0.2114	5.7691	0.2903
<b>5 Autovect.</b>	-	-	5.0565	0.2679	5.6867	0.2535	5.8187	0.3952
<b>10 Autovect.</b>	-	-	-	-	5.6868	0.6084	5.7068	0.4681
<b>15 Autovect.</b>	-	-	-	-	-	-	6.4158	0.5118

Tabla 5.3. Tiempo de cómputo consumido para el cálculo de las matrices de transformación de PCA y GPCA en función del número de imágenes y de autovectores considerados para la obtención del modelo de fondo.

### Calidad de las imágenes recuperadas

Finalmente, también se ha comparado el valor medio del error de recuperación calculado como la diferencia entre la imagen original y la recuperada al cuadrado. En la Figura 5.7 se muestra gráficamente el valor medio del error de recuperación (obtenido para 15 imágenes de fondo) en función del número de imágenes y el número de autovectores considerados en el cálculo del modelo de fondo. Se puede observar que en el caso de usar GPCA, el error de recuperación es inferior al de PCA, independientemente del número de imágenes y autovectores considerados.

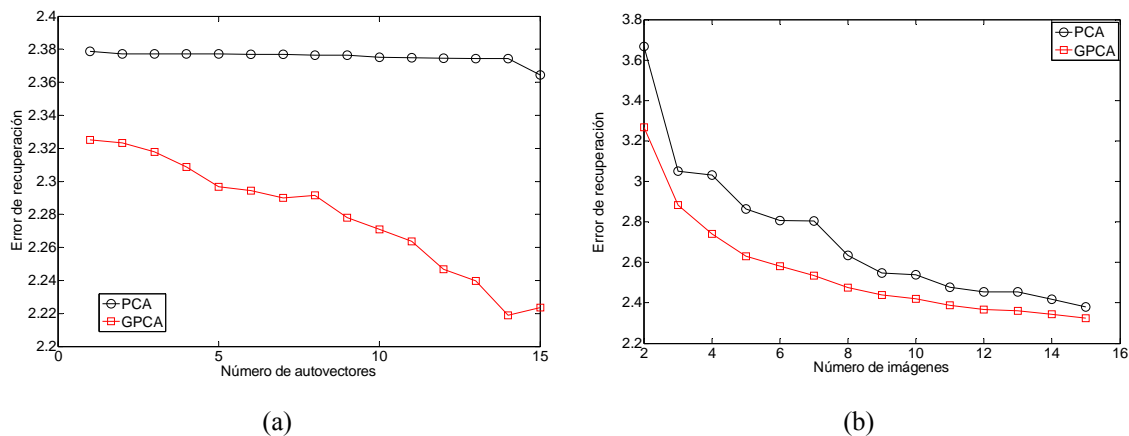


Figura 5.7. Error de recuperación de PCA y GPCA (a) En función del número de autovectores, para 15 imágenes de fondo. (b) En función del número de imágenes con 1 autovector.

A continuación, en la Tabla 5.4 se presentan los valores obtenidos para el error de recuperación medio, para diferentes valores del número de imágenes y autovectores considerados en el cálculo del modelo de fondo.

	2 Imágenes		5 Imágenes		10 Imágenes		15 Imágenes	
	PCA	GPCA	PCA	GPCA	PCA	GPCA	PCA	GPCA
1 Autovect.	3.6672	3.2688	2.8638	2.6307	2.54	2.4194	2.3789	2.3252
2 Autovect.	3.5395	3.2677	2.863	2.6295	2.538	2.4162	2.3772	2.3234
5 Autovect.	-	-	2.8204	2.618	2.5375	2.4004	2.3771	2.2966
10 Autovect.	-	-	-	-	2.5168	2.3622	2.3751	2.2709
15 Autovect.	-	-	-	-	-	-	2.3688	2.2235

Tabla 5.4. Error de recuperación medio en función del número de imágenes y autovectores considerados para la obtención del modelo de fondo. El error de recuperación se ha obtenido como la media del valor de recuperación medio obtenido para 15 imágenes de fondo.

Tras comparar los resultados obtenidos en las diferentes pruebas experimentales realizadas con PCA y GPCA se puede concluir que el uso de GPCA permite trabajar con matrices de dimensiones menores que en el caso de PCA, disminuyendo tanto los requerimientos de memoria como el tiempo de cómputo consumido en el cálculo de las matrices de transformación con respecto a PCA y proporcionando un menor error de recuperación a igual número de autovectores.

## 5.2. Selección de píxeles candidatos a pertenecer a robots móviles

En esta segunda etapa de la inicialización de las curvas que definen los contornos de los diferentes robots móviles, se compara cada imagen de entrada con el modelo de fondo obtenido en la etapa *off-line*, según se ha explicado en el apartado anterior. De esta forma es posible obtener un conjunto de píxeles de las imágenes de entrada que son candidatos a pertenecer a robots móviles presentes en la escena.

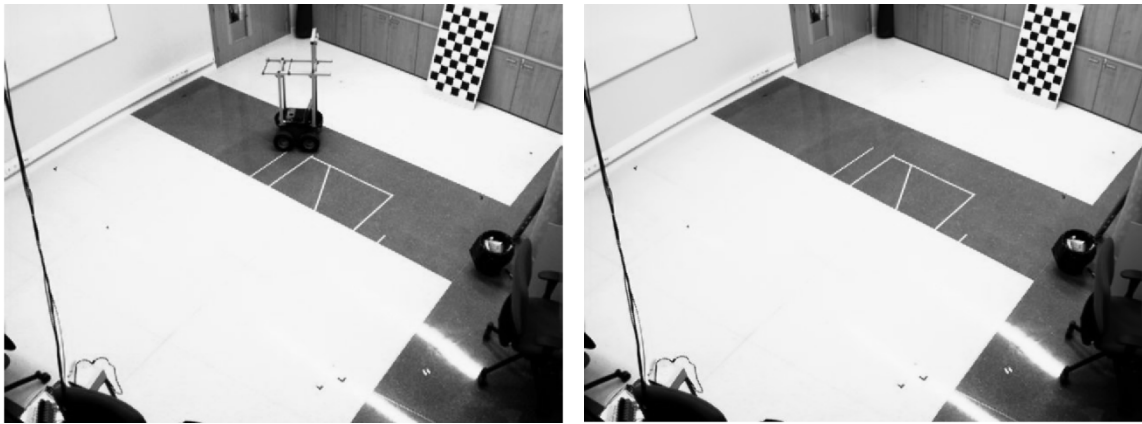
Como ya se ha comentado, el fondo de la escena se modela mediante dos matrices de transformación,  $L$  y  $R$ , obtenidas utilizando la técnica GPCA. Empleando estas dos matrices, cada imagen de entrada se proyecta al espacio transformado GPCA, de dimensión menor, mediante una transformación lineal definida según la ecuación (5.7) y a continuación, mediante la transformación inversa (5.8), se recupera la imagen en el espacio original.

$$I_T = \mathbf{L}^T (I - \mathbf{M})\mathbf{R} \quad (5.7)$$

$$\hat{I} = \mathbf{L}I_T\mathbf{R}^T + \mathbf{M} \quad (5.8)$$

Si la dimensión del espacio transformado es menor que la del espacio original (como ocurre en nuestro caso) en la transformación y posterior recuperación se pierde parte de la información contenida en la imagen original. Por este motivo la imagen recuperada no es exactamente igual que la original, de forma que aparece un error que se denomina error de recuperación y se define como la diferencia entre la imagen de entrada  $I_i$  y la imagen recuperada  $\hat{I}_i$ . El valor del error de recuperación será mayor en aquellos píxeles en los que exista una mayor diferencia entre la imagen de entrada y las imágenes utilizadas para la obtención del modelo de fondo.

En la Figura 5.8(b) se muestra una imagen de entrada en la que aparece un robot móvil. El resultado de proyectar esa imagen al espacio transformado GPCA de dimensión 1x1 (utilizando la ecuación (5.7)) y posteriormente recuperarla en el espacio original, (5.8), se muestra en la Figura 5.8(b). Se puede observar como los píxeles pertenecientes al fondo de la escena han sido recuperados correctamente. Sin embargo, los píxeles que corresponden al robot móvil no han podido ser recuperados debido a que las matrices de transformación GPCA ( $\mathbf{L}$  y  $\mathbf{R}$ ) únicamente contienen información del fondo de la escena.



(a) Imagen del robot (original)

(b) Imagen recuperada (1 autovalor)

Figura 5.8. (a) Imagen original en la que aparece un robot móvil (b) Imagen obtenida tras proyectar la imagen de entrada al espacio transformado (ecuación (5.7)) de dimensión 1x1 y posteriormente recuperarla en el espacio original (ecuación (5.8)).

Aplicando un umbral adecuado sobre el error de recuperación es posible discriminar entre los píxeles que pertenecen al fondo de la escena, y aquellos que corresponden a posibles robots móviles. Hay que tener presente que el valor del error de recuperación

será superior en los píxeles en los que la diferencia entre la imagen de fondo y la imagen de entrada sea mayor. Entre estos píxeles se encuentran los que pertenecen a robots móviles que hayan entrado en la escena después de la obtención del modelo de fondo de la misma.

En la Figura 5.9 se presentan, en forma de diagrama de bloques, las diferentes etapas necesarias para la selección de los píxeles candidatos a pertenecer a objetos en movimiento mediante la técnica GPCA. Cada una de las etapas representadas en el diagrama de bloques de la Figura 5.9 se describe en los siguientes apartados.

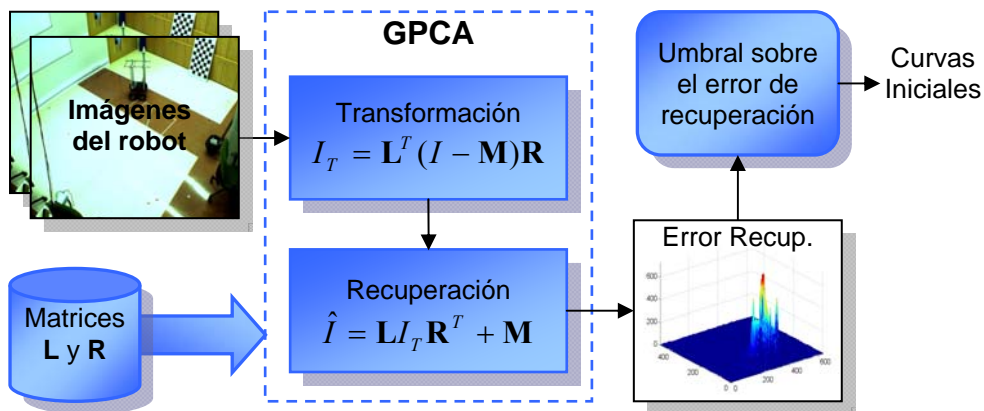


Figura 5.9. Diagrama de bloques para la selección de los píxeles candidatos a pertenecer a robots móviles mediante la umbralización del error de recuperación calculado utilizando la técnica GPCA.

### 5.2.1. Cálculo del error de recuperación

El error de recuperación puede obtenerse de forma directa como el valor absoluto de la diferencia entre la imagen original y la imagen recuperada, de forma que el valor del error en el píxel de coordenadas  $(w, i)$  se expresa según (5.9), en la que  $I_{wi}$  es el valor del píxel en la imagen original ( $I$ ) mientras que  $\hat{I}_{wi}$  es su valor en la imagen recuperada ( $\hat{I}$ ).

$$\varepsilon_{wi}^2 = \|I_{wi} - \hat{I}_{wi}\|^2 \quad (5.9)$$

Sin embargo, con objeto de aumentar la robustez frente al ruido, evitando que la presencia de ruido o brillos en las imágenes genere falsos candidatos, se define una ventana alrededor de cada píxel de la imagen y se obtiene el valor del error de recuperación para cada una de las ventanas.



Definiendo ventanas cuadradas de  $q \times q$  píxeles (siendo  $q > 1$  y siempre impar) para la imagen original y recuperada, el error de recuperación asociado al píxel central de la ventana cuyas coordenadas son  $(w, i)$  se obtiene mediante (5.10). En esta ecuación los valores de  $\Phi_{wi}$  y  $\hat{\Phi}_{wi}$  vienen dados por (5.11) y son el resultado de sumar los valores de los errores cuadráticos en todos los píxeles contenidos en la ventana definida sobre la imagen original y la recuperada, y dividir el resultado entre el número de píxeles de la ventana. Tanto en (5.9) como en (5.10) se puede observar que el error de recuperación se obtiene al cuadrado. De esta manera se elimina la necesidad de calcular raíces cuadradas para obtener la norma de la diferencia entre imágenes.

$$\varepsilon_{wi}^2 = \|\Phi_{wi} - \hat{\Phi}_{wi}\|^2 \quad (5.10)$$

$$\Phi_{wi} = \frac{1}{q^2} \sum_{j=w-\frac{q-1}{2}}^{w+\frac{q-1}{2}} \sum_{k=i-\frac{q-1}{2}}^{i+\frac{q-1}{2}} \mathbf{I}_{wi}, \quad \hat{\Phi}_{wi} = \frac{1}{q^2} \sum_{j=w-\frac{q-1}{2}}^{w+\frac{q-1}{2}} \sum_{k=i-\frac{q-1}{2}}^{i+\frac{q-1}{2}} \hat{\mathbf{I}}_{wi} \quad (5.11)$$

En la Figura 5.10 se muestra de forma gráfica el proceso de cálculo del error de recuperación, utilizando una ventana de tamaño  $3 \times 3$  píxeles ( $q = 3$ ).

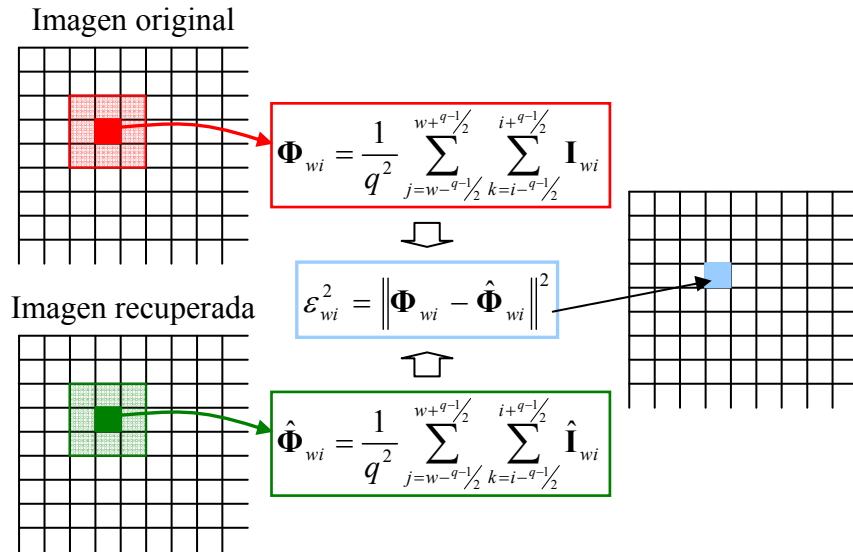


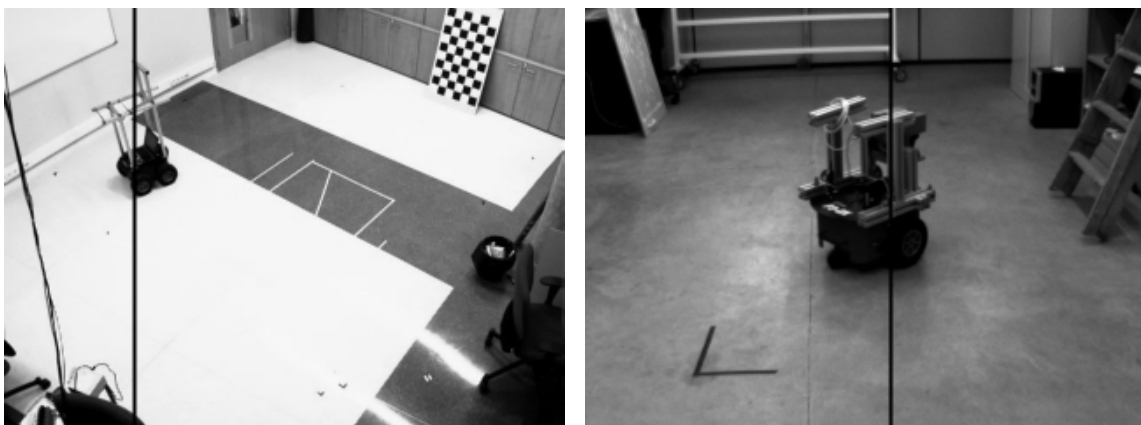
Figura 5.10. Cálculo del error de recuperación utilizando ventanas. En el caso de esta figura la ventana se ha considerado de  $3 \times 3$  píxeles.

A continuación se presentan algunas de las pruebas experimentales realizadas para determinar cómo afecta el hecho de utilizar ventanas al cálculo del error de recuperación. Además, se incluyen también diversas pruebas en las que se determina el efecto del tamaño de ventana elegido.

## Estudio del efecto del tamaño de la ventana utilizada en el cálculo del error de recuperación

Como ya se ha comentado, con objeto de aumentar la robustez frente al ruido, evitando que la presencia de ruido o brillos en las imágenes nos genere falsos candidatos, se ha decidido realizar el cálculo del error de recuperación utilizando ventanas cuadradas de tamaño  $qxq$  píxeles alrededor de cada píxel (según la ecuación (5.10)). El tamaño de ventana ( $q$ ) elegido influye tanto en el error de recuperación que se obtiene, como en el tiempo de cómputo consumido en el cálculo de dicho error. Para determinar cuál es el efecto de  $q$  sobre estos parámetros se han realizado diversas pruebas experimentales con diferentes imágenes pertenecientes a distintas secuencias que se presentan a continuación. Todos los experimentos mostrados en este apartado se han realizado utilizando las matrices de transformación GPCA utilizando 10 imágenes de fondo y un único autovector.

En primer lugar, en la Figura 5.11 se muestran dos de las imágenes sobre las que se han realizado los diferentes experimentos y para las que se presentarán los resultados obtenidos. En las dos imágenes de la Figura 5.11 se puede apreciar una línea negra, que se corresponde con una columna de la imagen (concretamente la columna 150 en la Figura 5.11(a) y la 350 en la Figura 5.11(b)). Estas columnas se han señalado debido a que en este apartado, además de representar el error de recuperación para toda la imagen, también se ha representado dicho error para una columna de forma que sea más sencillo apreciar las diferencias entre los resultados obtenidos en diferentes condiciones.



(a) Imagen perteneciente a la secuencia 1

(b) Imagen perteneciente a la secuencia 2

Figura 5.11. Imágenes pertenecientes a las secuencias con las que se han realizado las diferentes pruebas que permiten la evaluación del efecto del tamaño de la ventana utilizado en el cálculo del error de recuperación.

A continuación, en la Figura 5.12.a se muestra el error de recuperación obtenido como la diferencia entre la imagen de entrada y la imagen recuperada (ecuación (5.9)). Por otro lado, en la Figura 5.12 (b), (c) y (d) se presenta, de nuevo, el error de recuperación, que en este caso se ha calculado utilizando la expresión de la ecuación (5.10) con tamaños de ventana de  $q=3$ , 7 y 13 píxeles, respectivamente. Todas las gráficas mostradas en esta figura se han obtenido para la imagen de la secuencia 1 presentada en la Figura 5.11(a).

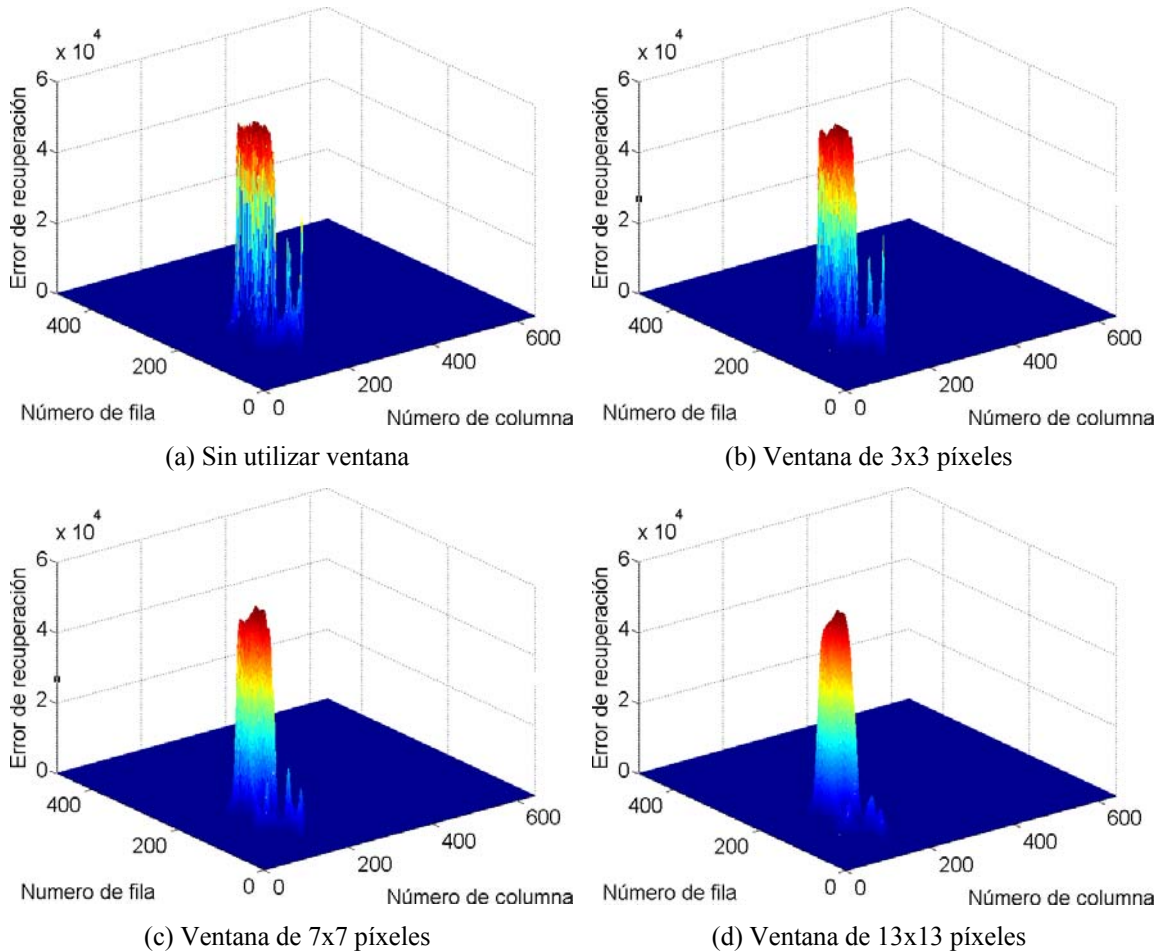


Figura 5.12. Error de recuperación en función del tamaño de la ventana utilizada para la imagen de la Figura 5.11.a. (a) Sin utilizar ventanas (ecuación (5.9)). (b) Con una ventana de 3x3 píxeles. (c) Con una ventana de 7x7 píxeles. (d) Con una ventana de 13x13 píxeles.

En la Figura 5.13 se muestra de nuevo el error de recuperación en las mismas condiciones expuestas para la Figura 5.12 pero, en este caso, para la imagen de la secuencia 2 mostrada en la Figura 5.11.(b).

Tanto en la Figura 5.12, como en la Figura 5.13 se puede observar como a medida que se aumenta el tamaño de la ventana, se produce un suavizado del error de recuperación.

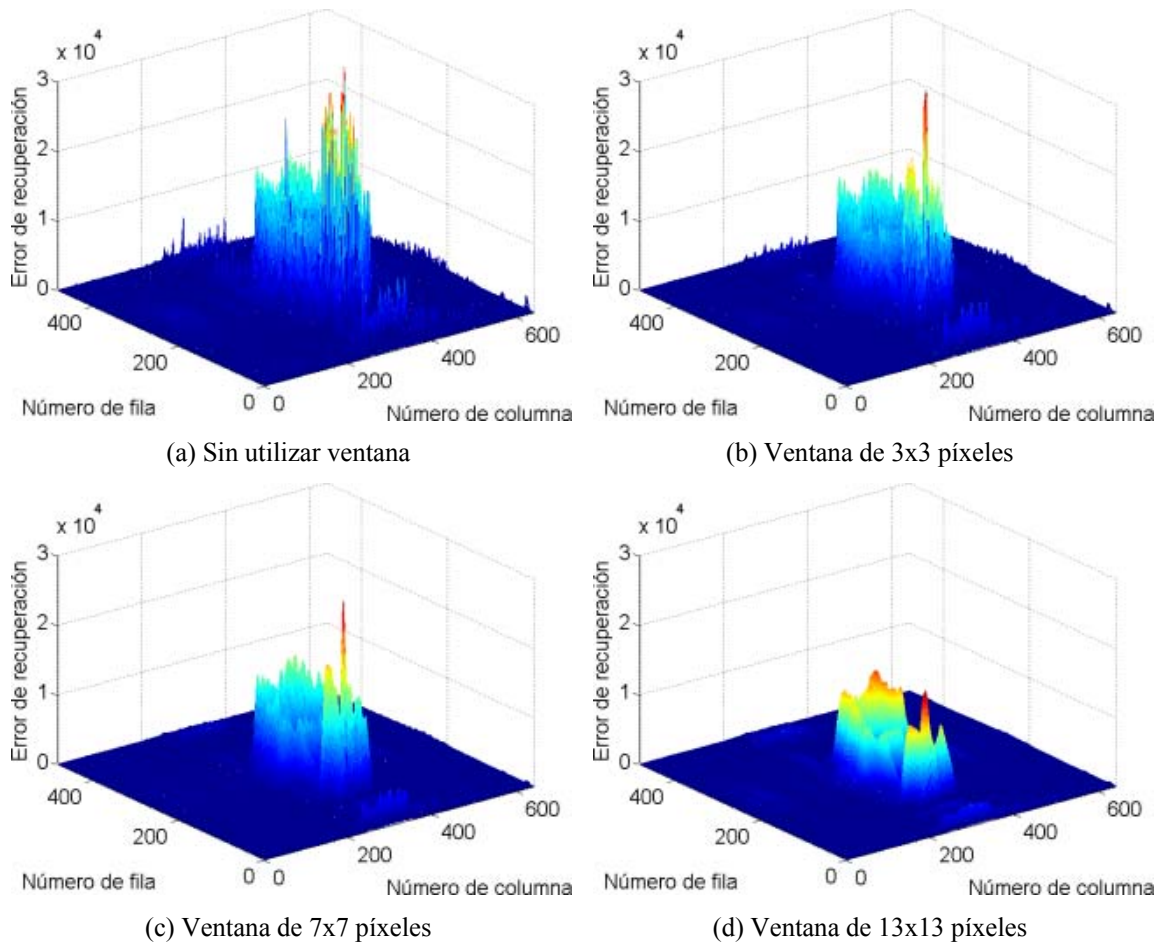


Figura 5.13. Error de recuperación en función del tamaño de la ventana utilizada, para la imagen de la Figura 5.11.b. (a) Sin utilizar ventanas (ecuación (5.9)). (b) Con una ventana de 3x3 píxeles. (c) Con una ventana de 7x7 píxeles. (d) Con utilizar una ventana de 13x13 píxeles.

El mismo efecto puede observarse también en la Figura 5.14 y la Figura 5.15 en la que se ha representado el valor del error de recuperación para los píxeles de una columna de cada una de las imágenes de la Figura 5.11.

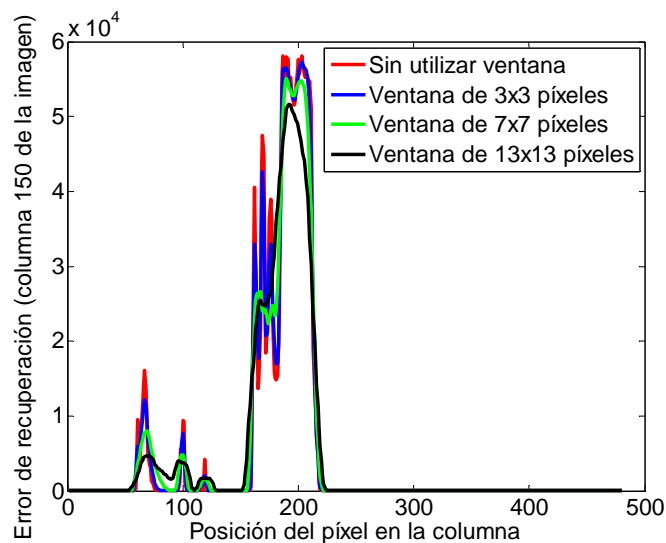


Figura 5.14. Valor del error de recuperación en los píxeles de la columna 150 de la imagen mostrada en la Figura 5.11.a, para diferentes valores del tamaño de ventana.

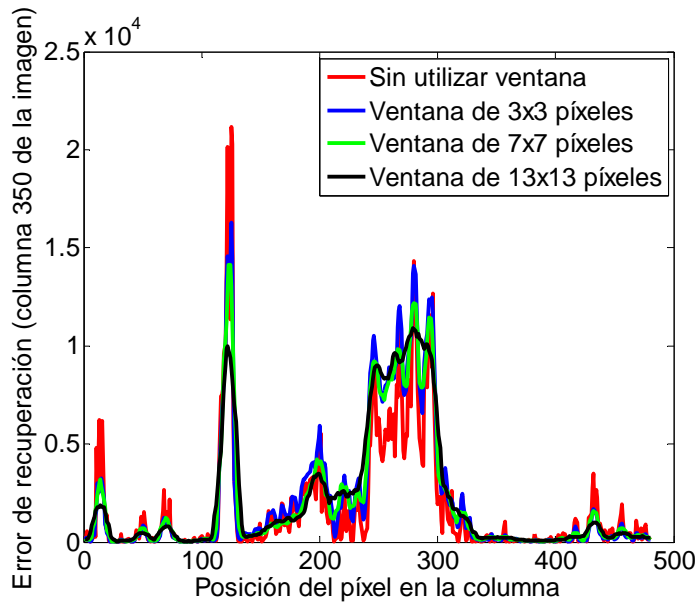


Figura 5.15. Valor del error de recuperación en los píxeles de la columna 350 de la imagen mostrada en la Figura 5.11 (b) para diferentes tamaños de ventana.

Respecto al tiempo de cómputo consumido en el cálculo del error de recuperación, en la Figura 5.16 se ha representado dicho tiempo (en segundos) en función del tamaño de la ventana utilizado. El valor mostrado en la Figura 5.16 se ha obtenido como el valor medio de los tiempos que se han calculado para 50 imágenes de cada una de las 2 secuencias (a las que corresponden las imágenes mostradas en la Figura 5.11). Se puede observar que, tanto en el caso de la secuencia 1 (Figura 5.11.a) como en el de la secuencia 2 (Figura 5.11.b) a medida que aumenta el tamaño de ventana, también se incrementa, de forma proporcional, el tiempo de cómputo.

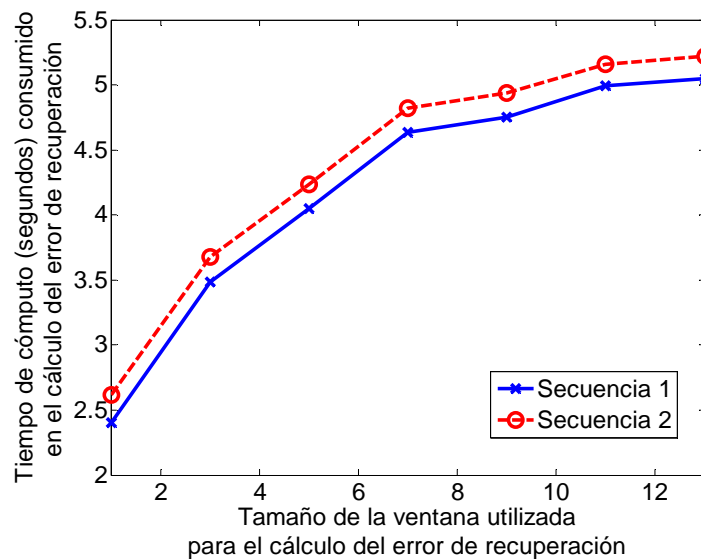


Figura 5.16. Tiempo de cómputo consumido en el cálculo del error de recuperación (en segundos) en función del tamaño de la ventana utilizada. En azul se muestra el tiempo consumido por 50 imágenes de la secuencia 1 (Figura 5.11.a). Con línea discontinua roja se presenta el tiempo de cómputo de las imágenes de la secuencia 2 (Figura 5.11.b).

Observando los diferentes resultados presentados en este apartado se aprecia que, a medida que se aumenta el tamaño de la ventana utilizada en el cálculo del error de recuperación, además de reducirse el efecto del ruido en las imágenes, se suaviza el error de recuperación. Además disminuye la diferencia del error de recuperación entre los píxeles que pertenecen a robots móviles, y aquellos que corresponden al fondo de la escena. Por otro lado, el aumento del tamaño de ventana, también provoca un incremento del tiempo de cómputo necesario para el cálculo del error de recuperación. Tras las pruebas experimentales realizadas, se ha decidido utilizar un tamaño de ventana de 3x3 píxeles para el cálculo del error de recuperación. Este tamaño de ventana es suficiente para reducir el ruido presente en las imágenes de entrada, sin incrementar excesivamente el tiempo necesario para el cálculo del error de recuperación.

### **5.2.2. Umbral del error de recuperación**

Una vez calculado el error de recuperación para cada imagen de entrada, es necesario aplicar un umbral a ese error para determinar qué píxeles son candidatos a pertenecer a robots móviles y cuáles corresponden al fondo de la escena. Se consideran píxeles candidatos aquellos que superan el valor del umbral elegido.

Es importante elegir correctamente el valor del umbral, ya que éste determina la calidad de las curvas iniciales. En el caso de elegir un umbral demasiado alto, se puede dar el caso de que píxeles pertenecientes a los robots se puedan confundir con los del fondo. Por el contrario, un umbral demasiado bajo puede provocar que píxeles pertenecientes al fondo se incluyan dentro de las curvas iniciales como parte de los robots móviles.

En la Figura 5.17 se muestra una imagen en la que aparece un robot móvil y sobre el mismo se muestran las curvas iniciales obtenidas tras aplicar diferentes umbrales al error de recuperación. En la Figura 5.17.a se muestra el resultado de aplicar un umbral demasiado elevado. Se puede observar como una parte importante del robot queda fuera de las curvas iniciales. Por otro lado, en la Figura 5.17.b se presentan las curvas obtenidas para un umbral cuyo valor es demasiado pequeño. En este caso, las curvas iniciales incluyen un gran número de píxeles pertenecientes al fondo de la escena. Finalmente, en la Figura 5.17.c se muestran las curvas tras ajustar correctamente el valor del umbral.

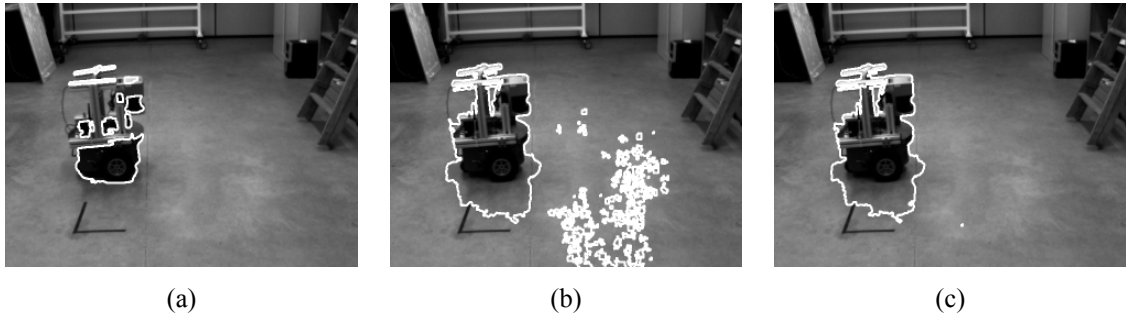


Figura 5.17. Curvas iniciales para diferentes valores del umbral sobre el error de recuperación (a) Curva obtenida con un umbral demasiado alto. (b) Curva obtenida con un umbral demasiado bajo. (c) Curva obtenida con el umbral adecuado.

### 5.2.2.1. Umbral proporcional al valor máximo del error de recuperación

Debido a que el valor del umbral está directamente relacionado con el error de recuperación, se propone calcular un valor inicial para el umbral en función del citado error, calculado para las imágenes de fondo de la escena. Posteriormente se presentan algunas alternativas que permiten actualizar el valor del umbral en función de la intensidad media de las imágenes de entrada. Con ello se consigue aumentar la robustez del algoritmo frente a cambios en la iluminación de la escena.

Bajo el supuesto de que el error de recuperación, definido según (5.10), debe tomar valores similares en los píxeles que pertenecen al fondo de la escena (tanto en imágenes con solo fondo como con presencia de robots) se propone calcular un valor inicial del umbral. En nuestro caso se propone utilizar un umbral inicial calculado a partir de las  $N_i$  imágenes utilizadas en la obtención del modelo de fondo.

Para ello, una vez obtenidas las matrices  $\mathbf{L}$  y  $\mathbf{R}$  que definen el modelo de fondo, cada una de las  $N_i$  imágenes utilizadas se proyecta (5.7), y se recupera, (5.8), para posteriormente calcular el error de recuperación. A continuación se obtiene el error de recuperación máximo de las imágenes de fondo, asignando a cada posición el valor máximo del error de recuperación en dicha posición para las  $N_i$  imágenes. Así, para imágenes de dimensiones  $f \times c$ , el error máximo viene dado por la ecuación (5.12), donde  $i$  y  $j$  representan respectivamente las filas y columnas de cada imagen.

$$\varepsilon_{r,fondo}^2(i, j) = \max\left(\left\{\varepsilon_{r,k}^2(i, j)\right\}_{k=1}^N\right) \quad i = 1, 2, \dots, f, \quad j = 1, 2, \dots, c \quad (5.12)$$

En la Figura 5.18 se muestran, en forma de diagrama de bloques, las diferentes etapas implicadas en el cálculo del valor inicial del umbral del error de recuperación.

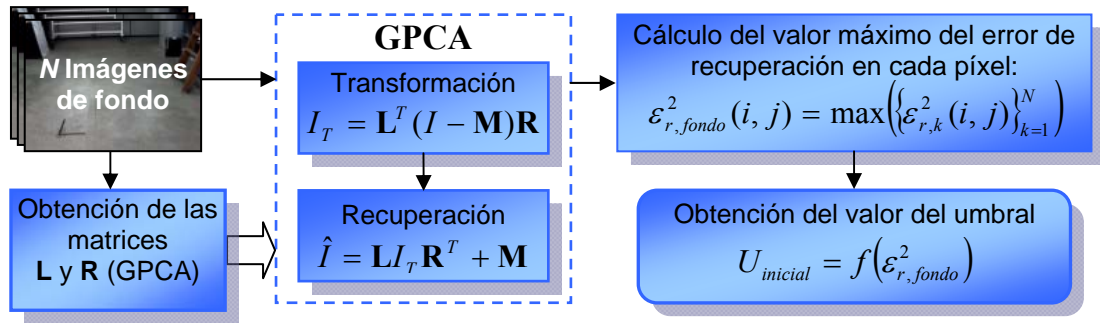


Figura 5.18. Diagrama de bloques general que refleja el proceso de obtención del valor inicial del umbral en función del valor máximo del error de recuperación obtenido para las imágenes implicadas en el cálculo del modelo de fondo.

Una vez obtenido el valor máximo del error de recuperación para cada píxel de las  $N_i$  imágenes de fondo, se define una función que relaciona el valor del umbral con dicho error de recuperación. En este trabajo se propone relacionar ambos parámetros mediante una constante de proporcionalidad, contemplándose dos posibles soluciones.

La primera alternativa plantea inicializar el umbral con un valor diferente para cada píxel de la imagen. Esto se hace asignando a cada píxel un umbral igual a una constante  $K$  por el valor máximo del error de recuperación de las imágenes de fondo en ese píxel, tal como se expresa en la ecuación (5.13).

$$U_{inicial}(i, j) = K \cdot \epsilon_{r,fondo}^2(i, j) \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n \quad (5.13)$$

Se han realizado diversas pruebas experimentales para comprobar la validez de esta alternativa. En estas pruebas se ha comprobado que la presencia de ruido en las imágenes hace que algunos de los píxeles que pertenecen al fondo se consideren como parte del robot. Esto se puede apreciar en las imágenes de la Figura 5.19.

Debido a que la función mostrada en (5.13) no proporciona buenos resultados en imágenes en las que se tiene ruido o cambios en las condiciones de iluminación, se propone una segunda alternativa definida en (5.14). Este umbral, al que denominamos umbral proporcional con  $K$  constante ( $U_p$ ), es igual para todos los píxeles de la imagen y se calcula como una constante  $K$  por el valor máximo del error de recuperación para todos los píxeles de la imagen.

$$U_p = K \cdot \max\left(\left\{\epsilon_{r,fondo}^2(i, j)\right\}_{\substack{i=1,\dots,n \\ j=1,\dots,m}}\right) \quad (5.14)$$



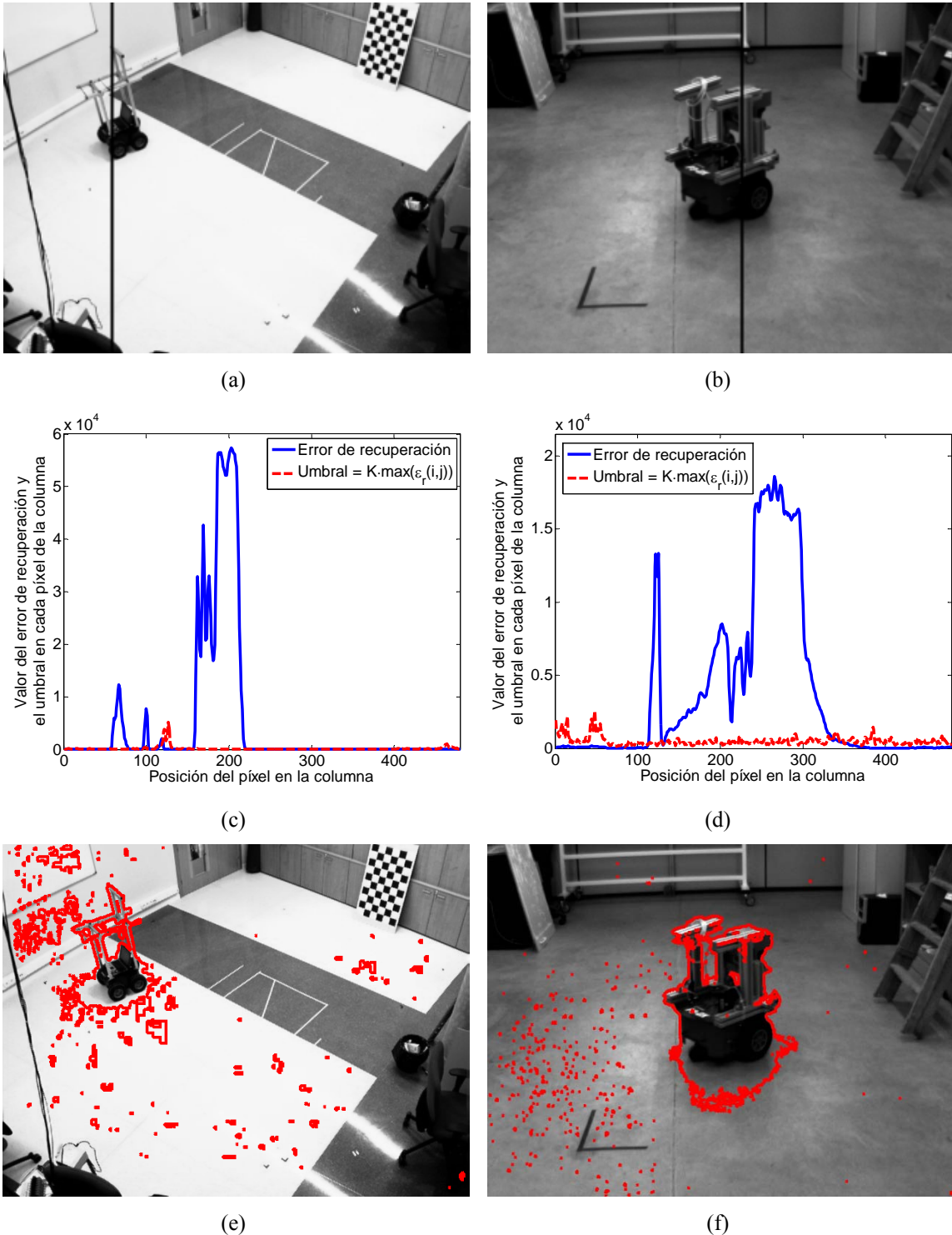


Figura 5.19. (a) y (b) Imágenes de entrada. (c) Valor del error de recuperación en los píxeles de la columna  $j=150$  marcada en negro en (a) y umbral proporcional (5.13). (d) Valor del error de recuperación en los píxeles de la columna  $j=350$  marcada en negro en (b) y umbral proporcional (5.13). (e) y (f) Curvas iniciales obtenidas tras aplicar el umbral proporcional definido en (5.13).

En la Figura 5.20 (a) y (b) se presentan dos imágenes que forman parte de dos secuencias diferentes. Si representamos el valor del error de recuperación de los píxeles que pertenecen a la columna marcada en color negro en las imágenes, se obtienen las gráficas presentadas en la Figura 5.20 (c) y (d).

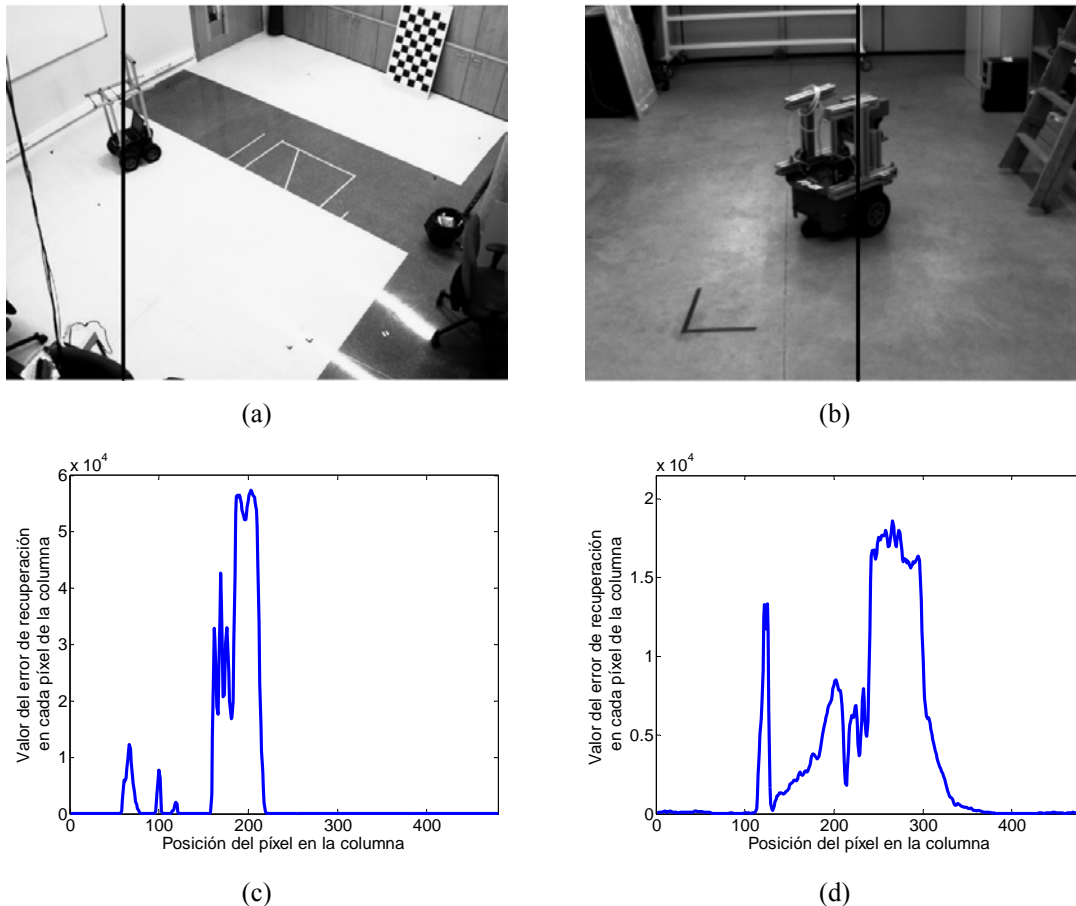


Figura 5.20. (a) y (b) Imágenes de entrada. (c) Valor del error de recuperación en los píxeles de la columna  $j=150$  marcada en negro en (a). (d) Valor del error de recuperación en los píxeles de la columna  $j=350$  marcada en negro en (b).

En las diferentes gráficas de la Figura 5.20 es evidente, a simple vista (para la mayor parte de los píxeles de las columnas consideradas) la diferencia entre los píxeles del fondo y los que forman parte del robot. Sin embargo, también se puede observar como existen diferencias significativas entre los valores que adquiere el error de recuperación en los diferentes píxeles, a pesar de que todos ellos pertenezcan al mismo robot móvil.

Para obtener un umbral inicial adecuado es importante ajustar correctamente el valor de la constante de proporcionalidad  $K$ , de forma que el valor inicial del umbral permita discriminar los píxeles que pertenecen al fondo de la escena de aquellos que corresponden a robots móviles.

Con objeto de determinar el valor más adecuado para la constante  $K$  se ha realizado un conjunto de pruebas experimentales, utilizando para ello diferentes secuencias de imágenes.

Para comenzar se realiza un ajuste inicial del valor de la constante  $K$ . Dicho ajuste es llevado a cabo de forma “manual” por un experto. Tal como se expone a continuación.

### **Ajuste inicial del valor de la constante de proporcionalidad del umbral del error de recuperación**

Para el ajuste del valor de la constante  $K$ , en primer lugar se calcula el error de recuperación de cada imagen mediante la expresión (5.10), considerando una ventana de  $3 \times 3$  píxeles.

A continuación se establece un umbral (definido según (5.14)) para el error de recuperación calculado en cada píxel. Este umbral dependerá de una constante  $K$  cuyo valor es ajustado de forma “manual” por un experto.

Se considera que un valor de la constante  $K$  es el mejor si al aplicar el umbral resultante, (5.14), al error de recuperación, las curvas obtenidas contienen la mayor parte posible de los píxeles pertenecientes a los robots móviles, sin incluir píxeles del fondo. En este caso, los píxeles que pertenecen a zonas de sombra proyectadas por los robots no se consideran píxeles de fondo. Por este motivo las curvas iniciales se consideran adecuadas aunque incluyan estos píxeles de sombra.

Hay que tener en cuenta que este proceso de umbralización asistida permite la obtención del umbral más adecuado para cada imagen de entrada. Sin embargo, esto no garantiza que el resultado de la umbralización coincida con el contorno de los robots móviles. Únicamente garantiza que las curvas son las mejores que es posible obtener aplicando únicamente un umbral al error de recuperación calculado para todos los píxeles de la imagen de entrada, sin ningún tipo de procesamiento previo ni posterior a la umbralización.

En la Tabla 5.5 se presentan los valores de la constante  $K$  para la que se han obtenido las mejores curvas iniciales, aplicando únicamente el umbral definido según la ecuación (5.14). Además, en la Tabla 5.5 también se muestra el valor del umbral utilizado. Todos los datos se han obtenido tanto para las imágenes de la secuencia 1 (Figura 5.11(a)) como para las de la secuencia 2 (Figura 5.11(b)).

Imagen	Secuencia 1 (Figura 5.11 (a))		Secuencia 2 (Figura 5.11 (b))	
	Valor de $K$	Umbral	Valor de $K$	Umbral
100	6.15	598.95	5.1	437.84
120	6.55	637.90	4.8	412.08
150	4.70	457.73	4.35	373.45
200	5.05	491.82	1.75	150.24
250	9.95	969.02	3.70	317.65
300	1.95	189.91	9.05	776.94
350	8.70	847.29	4.15	356.28
400	2.60	253.19	9.7	832.75
450	1.95	189.91	2.15	184.58
500	4.75	462.56	4.35	373.45

Tabla 5.5. Valor de la constante  $K$  óptima, elegida manualmente para el cálculo del umbral utilizando la ecuación (5.14) y valor del umbral para un conjunto de imágenes de prueba pertenecientes a las secuencias de las imágenes representadas en la Figura 5.11 (a) y (b).

A modo de ejemplo, en la Figura 5.21 se muestran dos de las imágenes de prueba utilizadas, concretamente la imagen 120 de ambas secuencias, junto con el resultado de la umbralización.



(a) Imagen de entrada (secuencia 1)



(b) Resultado de la umbralización



(c) Imagen de entrada (secuencia 2)



(d) Resultado de la umbralización

Figura 5.21. (a) Imagen de prueba de la secuencia 1. (b) Resultado obtenido tras umbralizar el error de recuperación de (a) con el valor del umbral óptimo. (c) Imagen de prueba de la secuencia 2. (d) Resultado obtenido tras umbralizar (c) con el valor del umbral óptimo.

Los valores obtenidos en la Tabla 5.5 permiten conocer el orden de magnitud de la constante de proporcionalidad  $K$ . Sin embargo, dada la diversidad de los valores obtenidos para las diferentes imágenes, no permiten obtener un valor de  $K$  válido para todas las imágenes de las secuencias de test. Por este motivo, es necesario realizar un estudio más exhaustivo para establecer el valor de la constante  $K$ .

A la vista de las imágenes de la Figura 5.20 (c) y (d), y los valores mostrados en la Tabla 5.5 se ha decidido definir un rango de valores de la constante  $K$  entre 2 y 25. En concreto se han realizado pruebas experimentales para los valores de  $K = [2, 5, 10, 25]$ . En la Figura 5.22 (a) y (c) se muestra, de nuevo, el valor del error de recuperación para una columna de las imágenes de prueba, junto con los diferentes valores obtenidos para el umbral que se aplica sobre dicho error. Las imágenes de la columna derecha de la Figura 5.22(b) y (d) muestran las curvas iniciales obtenidas al aplicar cada uno de los umbrales propuestos.

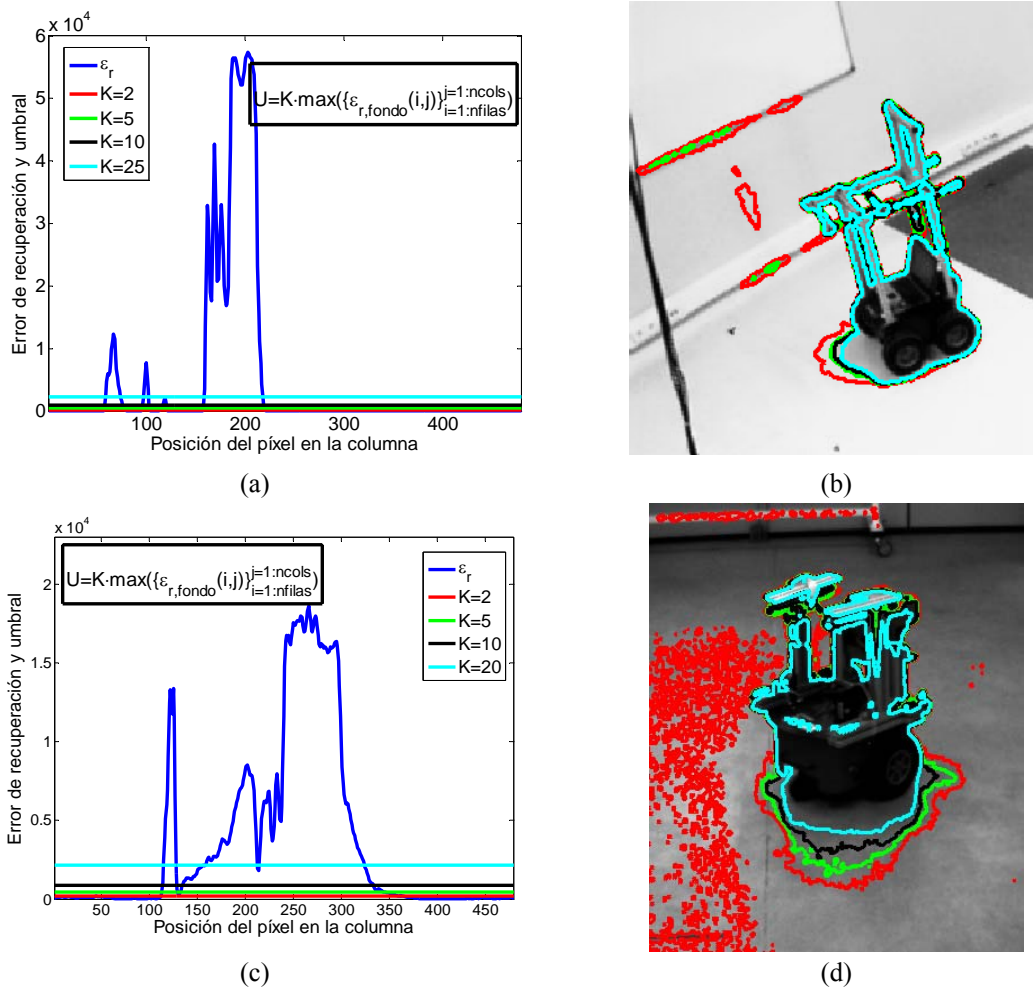


Figura 5.22. (a) y (c) Valor del error de recuperación para los píxeles de una columna de las imágenes mostradas en la Figura 5.20 (a) y (b) y umbrales para los diferentes valores de la constante  $K$  considerados. (c) y (d) Detalle de la imagen y las curvas iniciales obtenidas para los diferentes valores de la constante  $K$ .

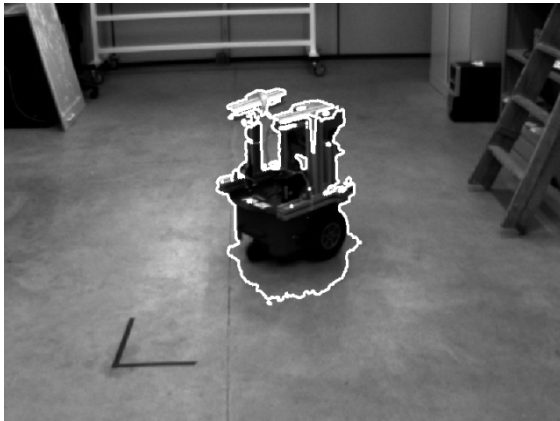
La primera conclusión que se puede extraer de estas imágenes es que eligiendo como umbral el doble del valor máximo del error de recuperación de las imágenes de fondo (en color rojo en las imágenes de la Figura 5.22) no es posible discriminar correctamente entre el fondo y el robot. Esto es debido a que, según se puede apreciar tanto en la Figura 5.22 (b) como en la Figura 5.22 (d), en las imágenes en que se incluye el robot existen zonas del fondo en las que el error de recuperación es superior al umbral impuesto, de forma que aparecen como parte del robot. Además, en el caso de la Figura 5.22 (b), el segundo umbral elegido según la ecuación (5.14) con  $K = 5$  tampoco es suficientemente grande como para eliminar todos los píxeles del fondo. En el otro extremo, si se comprueban las curvas obtenidas al aplicar un umbral utilizando la ecuación (5.14) con  $K = 25$ , en ambas figuras se puede observar como se han eliminado de las curvas todos los píxeles que pertenecen al fondo (excepto los debidos a las sombras), pero también se han eliminado algunos de los píxeles que forman parte del robot.

A la vista de los resultados obtenidos, se fija el valor del umbral en:  $U_p = 10 \cdot \max\left(\{\mathcal{E}_{r,fondo}^2(i,j)\}_{i=1,\dots,m}^{j=1,\dots,n}\right)$  ya que, según se puede observar, es el valor más adecuado para todas las pruebas experimentales realizadas.

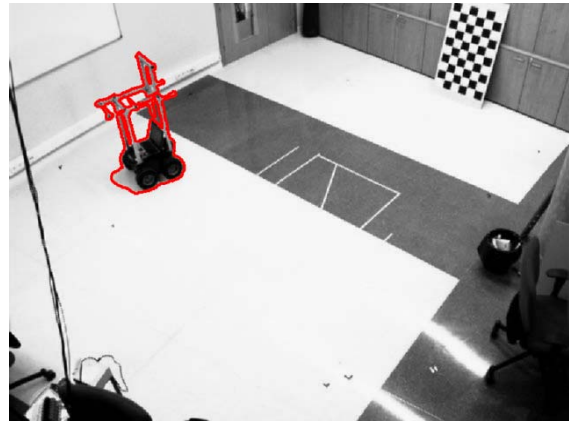
Para evaluar la validez del umbral proporcional con  $K=10$ , se ha aplicado el mismo al error de recuperación de diferentes imágenes pertenecientes a dos secuencias adquiridas en el ISPACE-UAH. Las curvas iniciales obtenidas en cada caso se presentan en la Figura 5.23. Se puede observar que el umbral obtenido permite diferenciar, en todas las imágenes, los píxeles que corresponden a los robots móviles de los que pertenecen al fondo.

Las diferentes imágenes de la Figura 5.23 permiten evaluar de forma cualitativa las curvas iniciales obtenidas en el caso de utilizar un umbral definido según la ecuación (5.14) con  $K=10$ . Sin embargo, también es deseable contar con una evaluación cuantitativa de estas curvas. Para ello se comparan, píxel a píxel, las curvas obtenidas al aplicar el umbral proporcional (5.14) con  $K=10$  con las obtenidas de forma manual con los valores de  $K$  mostrados en la Tabla 5.5 .

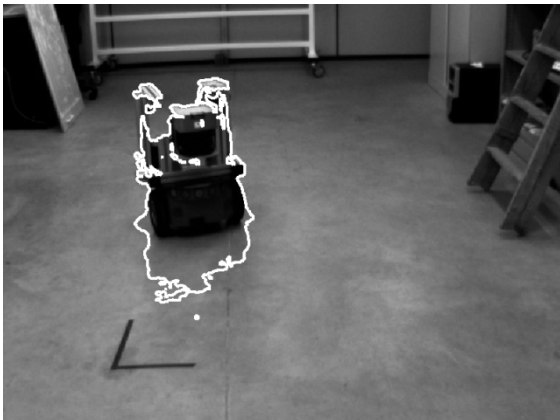




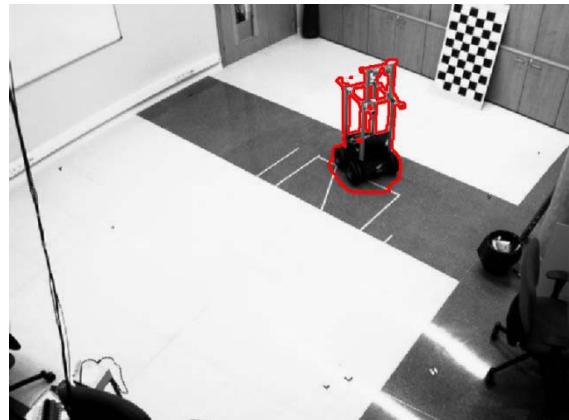
(a) Imagen 120



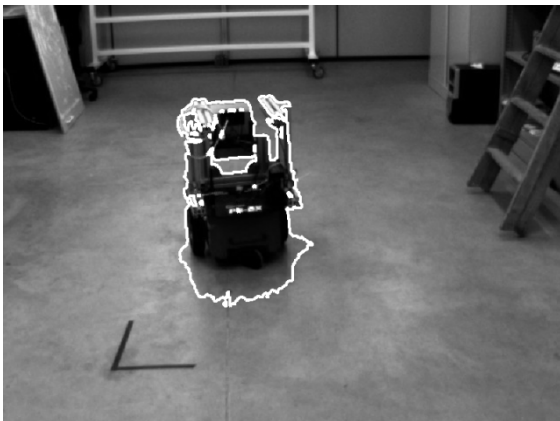
(b) Imagen 120



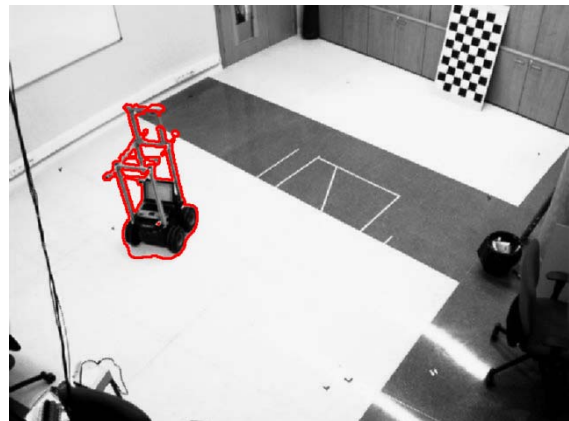
(c) Imagen 250



(d) Imagen 250



(e) Imagen 400



(f) Imagen 400

Figura 5.23. Curvas iniciales obtenidas para diferentes imágenes al aplicar sobre el error de recuperación un umbral proporcional (con  $K=10$ ) al valor máximo del error de recuperación de las imágenes de fondo (5.14).

En la Tabla 5.6 se muestran, a modo de resumen, los resultados obtenidos tras comparar las imágenes segmentadas aplicando un umbral con  $K = 10$ , y aquellas en las que le valor de  $K$  ha sido ajustado manualmente para imágenes de las secuencias 1 y 2 (Figura 5.20 (a) y (c)), respectivamente.

En la primera columna de la Tabla 5.6 se presenta el porcentaje de píxeles de la imagen que han sido segmentados de forma correcta. Se puede observar que para todas las imágenes contempladas en esta prueba, el número de píxeles que han sido segmentados en ambas supera siempre el 97%. Por otro lado, en la segunda columna de la Tabla 5.6 se muestra, en forma de porcentaje, el número de píxeles que han sido considerados como parte del robot móvil al aplicar el umbral con  $K = 10$ , y no fueron segmentados en la imagen con que se compara. Finalmente, en la última columna de la Tabla 5.6 se presenta el porcentaje de píxeles que se consideraron como parte del robot móvil en la segmentación manual, y no han sido considerados al aplicar el umbral (ecuación (5.14)) con  $K=10$ .

Imagen		Aciertos	Falsos positivos	Píxeles sin segmentar
Secuencia 1	100	98.8851 %	0 %	1.1149 %
	150	97.5645 %	0 %	2.4355 %
	200	97.2907 %	0 %	2.7093 %
	250	98.5384 %	0 %	1.4616 %
	300	99.6807 %	0 %	0.3193 %
	350	98.0915 %	0 %	1.9085 %
	400	99.4264 %	0 %	0.5736 %
Imagen		Aciertos	Falsos positivos	Píxeles sin segmentar
Secuencia 2	100	99.9909 %	0.0091 %	0 %
	150	99.7786 %	0 %	0.2214 %
	200	99.9274 %	0 %	0.0726 %
	250	99.9414 %	0.0586 %	0 %
	300	99.6758 %	0 %	0.3242 %
	350	99.7738 %	0 %	0.2262 %
	400	99.8675 %	0 %	0.1325 %

Tabla 5.6. Porcentaje de píxeles segmentados correctamente, falsos positivos y píxeles pertenecientes a los robots sin segmentar para las imágenes de la secuencia 1 (Figura 5.20 (a)) y la secuencia 2 (Figura 5.20 (b)).

Los resultados presentados en la Tabla 5.6 nos permiten concluir que la constante de proporcionalidad elegida ( $K = 10$ ) proporciona una segmentación correcta de las imágenes de entrada de las secuencias mostradas en la Figura 5.11.

Como se ha visto, el umbral elegido (definido en (5.14)) es proporcional al valor máximo del error de recuperación de las  $N_i$  imágenes de fondo (ecuación (5.12)). Por



tanto, el valor del umbral dependerá de los mismos parámetros que ese error de recuperación. Estos parámetros son:

- el número de imágenes y autovalores considerados para la obtención del modelo de fondo,
- el tamaño de la ventana ( $q$ ) considerado en el cálculo del error de recuperación (ecuación (5.10)).

Para evaluar el efecto de estos parámetros y validar el valor establecido para la constante de proporcionalidad,  $K$ , se han realizado diversas pruebas experimentales que se presentan a continuación.

### **Efecto del número de imágenes y de autovectores utilizados para el cálculo del modelo de fondo sobre el umbral proporcional con $K$ constante**

Según se ha explicado en el apartado 5.1.1 el número de imágenes y autovectores utilizados para el cálculo del modelo de fondo influye tanto en el valor del error de recuperación, como en el tiempo de cómputo.

A continuación se presentan los resultados que muestran el efecto sobre el umbral proporcional con  $K$  constante de la variación del número de imágenes y autovectores. En todas las pruebas experimentales presentadas en este apartado el error de recuperación se ha obtenido utilizando la ecuación (5.10), considerando una ventana de 3x3 píxeles ( $q=3$ ) alrededor de cada píxel. En cuanto al valor del umbral, a la vista de los resultados anteriores se ha decidido fijar el valor de la constante  $K=10$ , es decir,  $U_p = 10 \cdot \max \left( \left\{ \varepsilon_{r, fondo}^2(i, j) \right\}_{\substack{i=1, \dots, n \\ j=1, \dots, m}} \right)$ .

En primer lugar, en la Figura 5.24 se muestra gráficamente el valor del umbral en función del número de imágenes de fondo considerado para 1, 2 y 5 autovectores, para las secuencias a las que pertenecen las imágenes de la Figura 5.20 (a) y (b). Se puede observar cómo el valor del umbral apenas varía a medida que aumenta el número de imágenes y de autovectores utilizados para obtener el modelo de fondo.

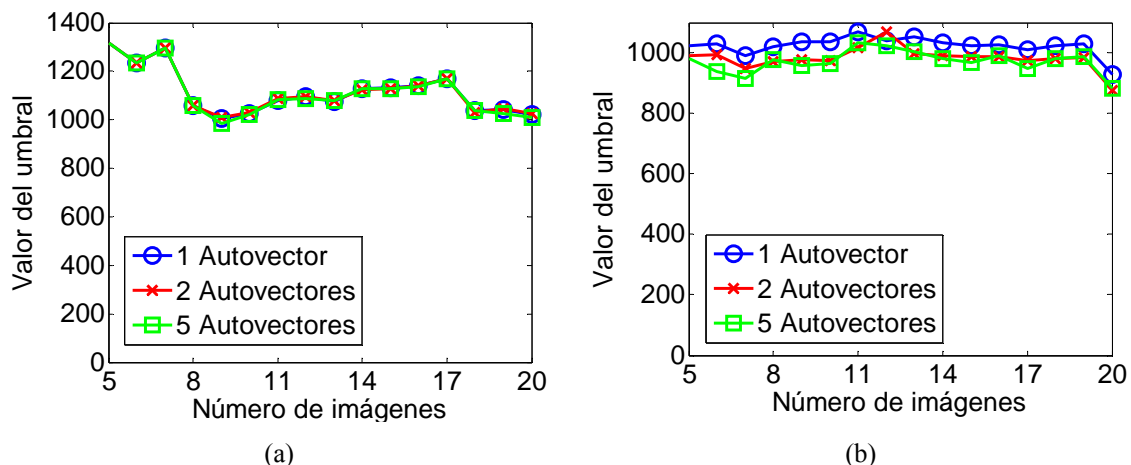


Figura 5.24. Valor del umbral proporcional, con  $K=10$ , en función del número de imágenes para 1, 2 y 5 autovectores. (a) Imágenes de la secuencia 1 (Figura 5.20 (a)). (b) Imágenes de la secuencia 2 (Figura 5.20 (b))

Con objeto de evaluar de forma conjunta el efecto de los parámetros bajo estudio (número de imágenes y de número de autovectores considerados en el cálculo del error de recuperación) en la Figura 5.25 (a) y (b) se muestran gráficamente los valores de los umbrales obtenidos en función de ambos parámetros, para una imagen de la secuencia 1 (Figura 5.20(a)) y otra de la secuencia 2 (Figura 5.20(b)). De nuevo, se puede observar como ninguno de los parámetros modifica de forma significativa el valor del umbral.

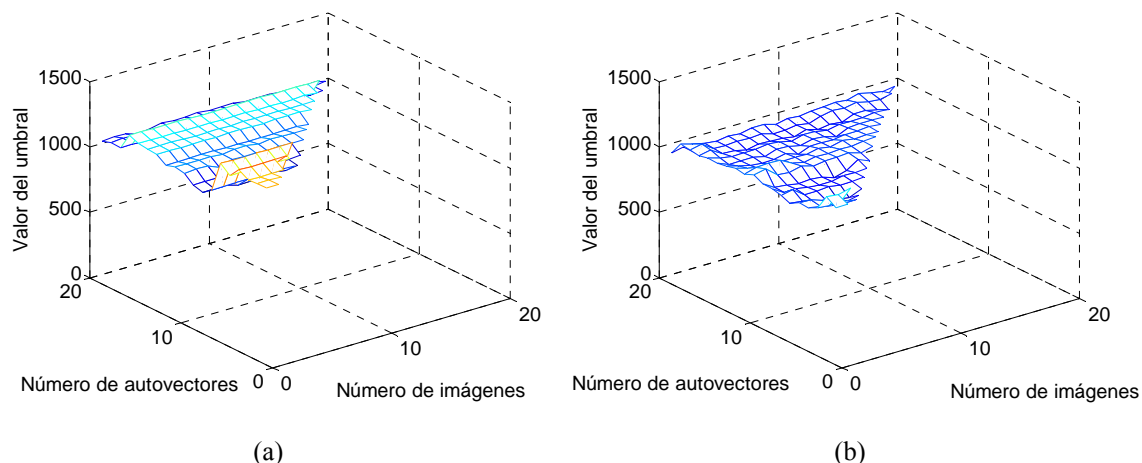


Figura 5.25. Valor del umbral proporcional (5.14) con  $K=10$  en función del número de imágenes y autovectores considerados en la obtención del modelo de fondo de la escena usando GPCA (a) para las imágenes de la secuencia 1, (b) para las imágenes de la secuencia 2.

Con respecto al tiempo de cómputo consumido en la obtención de las curvas iniciales, en la Figura 5.26 se muestran los tiempos consumidos en el cálculo y umbralización del error de recuperación. Para obtener estos tiempos, se ha realizado la inicialización de una secuencia de 50 imágenes y se ha representado el valor medio del tiempo consumido (en segundos) para los diferentes valores del número de imágenes y de autovectores considerados para el cálculo del modelo de fondo. El hardware utilizado

corresponde al del nodo cliente, que ha sido definido en el apartado 3.1.1, al describir la arquitectura del ISPACE-UAH.

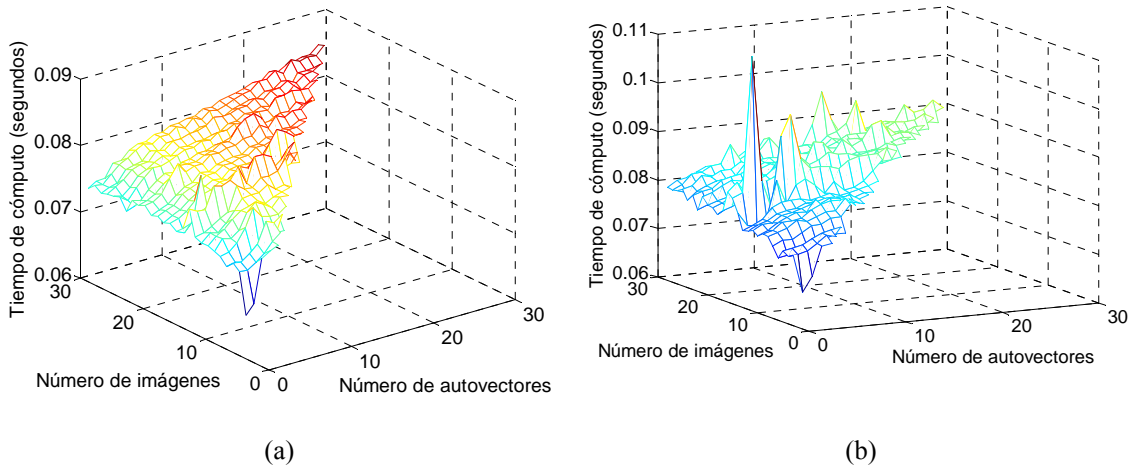


Figura 5.26. Tiempo de cómputo consumido en el cálculo y umbralización del error de recuperación para la obtención de las curvas iniciales en función del número de imágenes y de autovectores considerados en el modelado del fondo de la escena. (a) Secuencia de imágenes 1 (Figura 5.20 (a)). (b) Secuencia de imágenes 2 (Figura 5.20 (b)).

En la Figura 5.26 se observa que el tiempo de cómputo del modelo de fondo se incrementa tanto con el número de imágenes como con el número de autovectores, sin embargo las mejoras que se introducen en el cálculo del umbral no son apreciables a partir de  $N_i=10$  imágenes (según se puede apreciar en la Figura 5.25). Por este motivo, el número de imágenes para el cálculo del modelo de fondo y la obtención del valor inicial del umbral se fija en  $N_i=10$  imágenes y se utiliza un único autovalor. Cabe destacar que la conclusión de este apartado es la misma que se ha obtenido en el apartado 5.1.1 al estudiar el efecto del número de imágenes y autovectores sobre el modelo de fondo.

### Efecto del tamaño de la ventana sobre el umbral proporcional con $K$ constante

Como se ha visto en el apartado 5.2.1, el tamaño de la ventana utilizada para el cálculo del error de recuperación en cada píxel influye sobre el valor de ese error y, consecuentemente, también tiene influencia sobre el valor inicial del umbral.

En este apartado se estudia el efecto del tamaño de ventana sobre el valor inicial del umbral. Para ello se han realizado diferentes pruebas experimentales en las que se ha modificado el tamaño de la ventana, manteniendo fijos tanto el número de imágenes, como el de autovalores y fijando  $K=10$  en la ecuación (5.14). A continuación se presentan los resultados obtenidos.

En la Tabla 5.7 se muestran los valores iniciales del umbral en función del tamaño de ventana elegido para un valor fijo de la constante  $K=10$ . Se considera que la secuencia 1 es a la que pertenece la imagen de la Figura 5.20 (a) mientras que la secuencia 2 es la correspondiente a la imagen de la Figura 5.20 (b).

Tamaño de la ventana	Secuencia 1		Secuencia 2	
	Umbral	% Aciertos	Umbral	% Aciertos
$q = 1$	3802.3977	99.4616 %	3250.1043	97.1250 %
$q = 3$	906.35547	99.9808 %	867.2909	99.4486 %
$q = 5$	614.32094	99.7041 %	403.72259	99.1117 %
$q = 7$	516.71103	99.4466 %	253.07834	97.9326 %
$q = 9$	353.49215	99.1465 %	175.26719	83.2454 %
$q = 11$	254.52228	98.8796 %	160.08304	78.6615 %
$q = 13$	200.21191	98.6286 %	144.53761	74.9694 %
$q = 15$	179.59117	98.4082 %	130.6975	71.8835 %

Tabla 5.7. Valor inicial del umbral en función del tamaño de la ventana utilizada para el cálculo del modelo de fondo y número de píxeles segmentados correctamente (en porcentaje).

Si se representan gráficamente estos valores, se obtienen las curvas mostradas en la Figura 5.27.

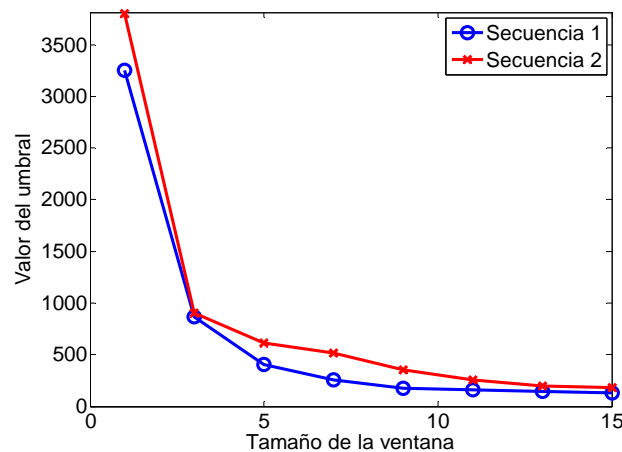


Figura 5.27. Valor del umbral en función del tamaño de la ventana utilizada para el cálculo del error de recuperación.

Tanto en la Tabla 5.7 como en la Figura 5.27 se puede apreciar como a medida que se incrementa el tamaño de la ventana, el valor del umbral disminuye. Esto se debe al hecho de que al emplear ventanas en el cálculo del error de recuperación se reducen los efectos del ruido en las imágenes, eliminando los picos aislados. Esto se puede ver en la Figura 5.28 en la que se ha representado el valor máximo del error de recuperación de las imágenes de fondo en cada píxel para diferentes valores del tamaño de ventana. En la primera imagen (Figura 5.28(a)) se puede observar que al realizar el cálculo del error

de recuperación, restando la imagen original y la recuperada pixel a pixel, aparecen picos dispersos debidos a la presencia de ruido en las imágenes. Mientras que, en la Figura 5.28(b), en la que se ha utilizado una ventana de 3x3 píxeles en el cálculo del error de recuperación, esos picos se han eliminado. Ambas imágenes se han representado con la misma escala en el eje Z para realizar la comparación.

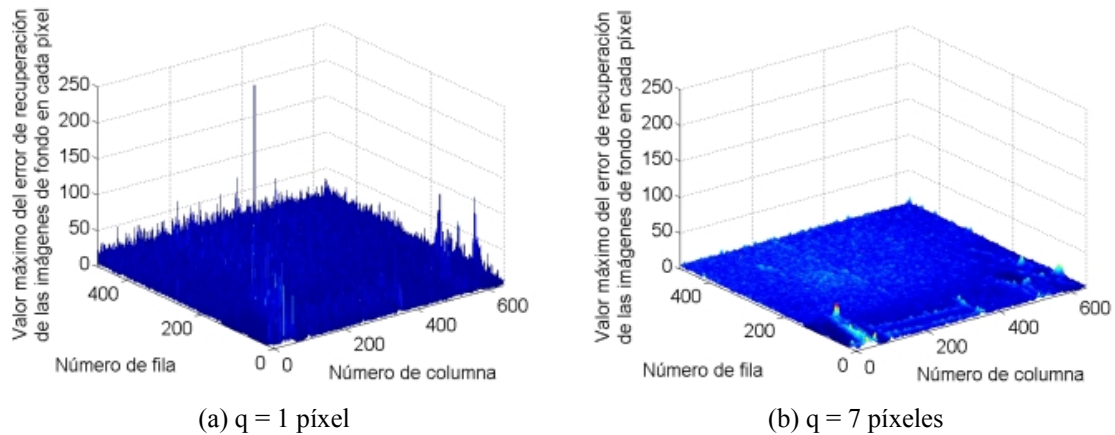


Figura 5.28. Valor máximo del error de recuperación de las imágenes de fondo en cada píxel para diferentes valores del tamaño de la ventana elegida (a)  $q = 1$  píxeles (b)  $q = 7$  píxeles.

En la Figura 5.29 se ha representado, de nuevo, el valor máximo del error de recuperación de todas las imágenes del fondo para los píxeles de una columna de la imagen. De nuevo se puede observar que a medida que se incrementa el tamaño de la ventana, se reduce la altura de los picos en el error de recuperación.

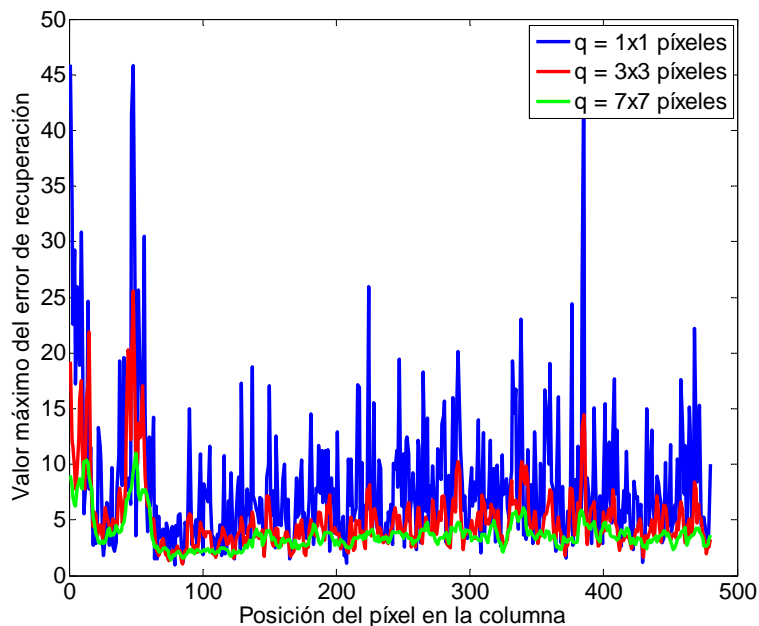
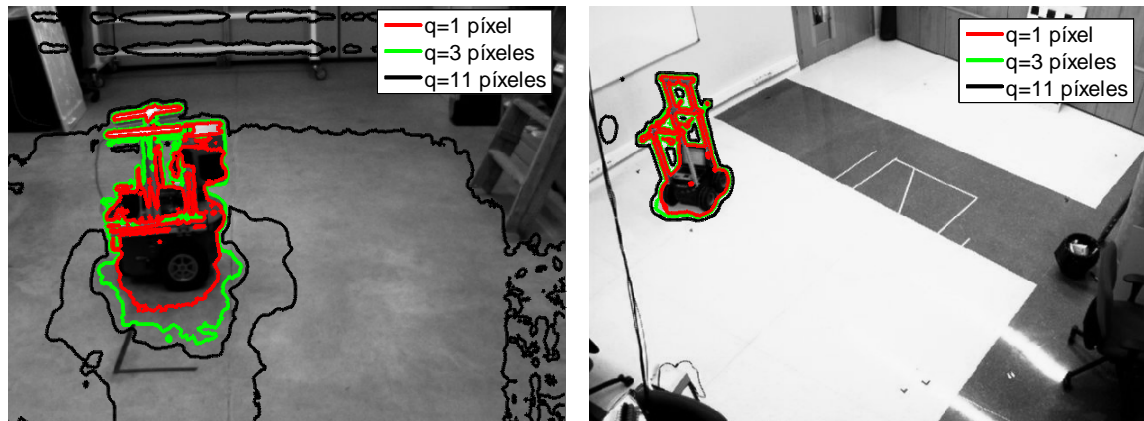


Figura 5.29. Valor máximo del error de recuperación de las imágenes de fondo en función del tamaño de la ventana para una columna de la imagen.

Esta reducción del valor del error de recuperación y, consecuentemente, del umbral, puede provocar que, si el tamaño elegido para la ventana es demasiado grande el umbral sea poco restrictivo. De esta forma, si las imágenes están contaminadas con ruido, o existen pequeños cambios en la iluminación, algunos píxeles que pertenecen al fondo pueden considerarse parte del robot.



(a) Imagen 100 de la secuencia 1

(b) Imagen 100 de la secuencia 2

Figura 5.30. Curvas iniciales obtenidas tras aplicar sobre el error de recuperación el umbral proporcional a  $K$  con  $K=10$  calculado para un tamaño de ventana de  $q = 1, 3$  y  $11$  píxeles.

Este efecto se observa en la Figura 5.30 donde se aprecia que las curvas obtenidas para un tamaño de ventana de  $11 \times 11$  píxeles ( $q=11$ ) incorporan píxeles del fondo a la curva que contiene el robot móvil. Esto implica que, en caso de que se aumente el tamaño de ventana ( $q$ ), el valor de la constante de proporcionalidad  $K$  también debe incrementarse para que el umbral proporcional (5.14) continúe siendo válido.

Con objeto de determinar los valores de la constante  $K$  adecuados para cada tamaño de ventana se compara el resultado de aplicar el umbral definido en la Tabla 5.5 para las imágenes de prueba con el resultado obtenido para diferentes valores de la constante  $K$  y del tamaño de ventana. En la Figura 5.31 se ha representado el número de píxeles que se encuentran dentro de la curva inicial en función del valor del valor de  $K$  empleado para la obtención del umbral dado por la ecuación (5.14) y del tamaño de ventana utilizada en el cálculo del error de recuperación. Además, se ha representado también el número de píxeles obtenidos en la segmentación manual de las imágenes.

La intersección entre la curva obtenida y el plano definido por la segmentación manual nos indicará qué rango de valores de la constante  $K$  es válido para cada uno de los valores del tamaño de ventana.

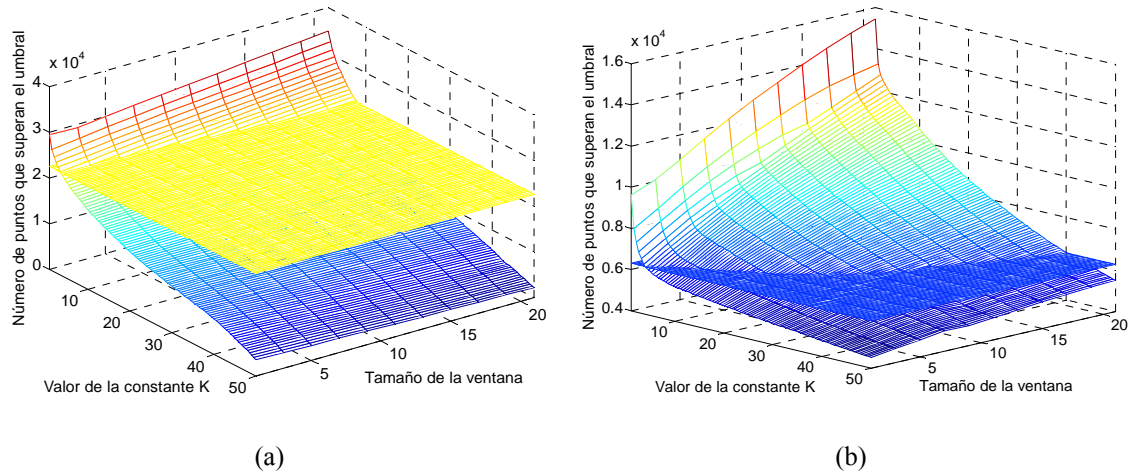


Figura 5.31. Número de píxeles en los que el error de recuperación supera el umbral (Ecuación (5.14)) en función del valor de la constante  $K$  y el tamaño de ventana utilizado en el cálculo del error. (a) Imagen de la secuencia 1. (b) Imagen de la secuencia 2.

Representando los valores de  $K$  en los que existe intersección entre la curva y el plano para cada tamaño de ventana se obtiene la gráfica mostrada en la Figura 5.32 donde se puede observar que, como cabía esperar, a medida que aumenta el tamaño de ventana, el valor de la constante  $K$  también debe ser mayor, con objeto de que el valor del umbral no disminuya de forma drástica.

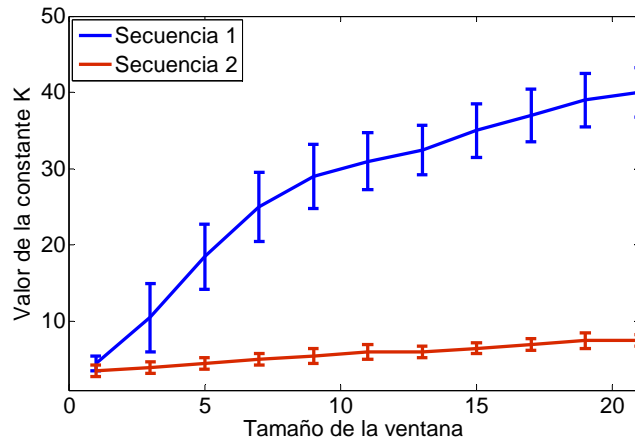


Figura 5.32. Valores válidos de la constante  $K$  en el cálculo del umbral sobre el error de recuperación en función del tamaño de la ventana utilizada para las imágenes de entrada mostradas en la Figura 5.19(a) y (b).

Las curvas obtenidas para ambas secuencias pueden ajustarse mediante una ecuación de segundo grado como la que se presenta en (5.15) para la secuencia 1 y en (5.16) para el caso de la secuencia 2. En ambos casos  $q$  es el tamaño de la ventana utilizada.

$$K = -0.0916 \cdot q^2 + 3.7001 \cdot q + 1.5069 \tag{5.15}$$

$$K = -0.0029 \cdot q^2 + 0.2686 \cdot q + 3.2413 \tag{5.16}$$



Si se calcula de nuevo el umbral modificando el valor de la  $K$  en función del tamaño de ventana utilizado, se obtienen los valores mostrados en la Tabla 5.8. En esta tabla se muestra además, en porcentaje, el número de píxeles segmentado correctamente con cada uno de los umbrales y tamaños de ventana presentados.

Tamaño ventana	Secuencia 1		Secuencia 2	
	Umbral	% Aciertos	Umbral	% Aciertos
$q = 1$	463.63568	98.2454 %	1139.8093	99.7145 %
$q = 3$	1067.9149	99.1995 %	1306.8601	99.7891 %
$q = 5$	1605.7492	99.5947 %	1466.3349	99.4453 %
$q = 7$	2077.1386	99.3783 %	1618.2338	99.2126 %
$q = 9$	2482.083	98.9469 %	1762.5566	98.8398 %
$q = 11$	2820.5825	98.8118 %	1899.3034	98.4531 %
$q = 13$	3092.637	98.6094 %	2028.4742	98.1032 %
$q = 15$	3298.2466	98.3649 %	2150.069	97.8919 %
$q = 17$	3437.4112	98.2865 %	2264.0878	97.6090 %
$q = 19$	3510.1309	98.0446 %	2370.5306	97.3245 %
$q = 21$	3516.4057	97.7829 %	2469.3974	97.0583 %

Tabla 5.8. Valor inicial del umbral en función del tamaño de la ventana utilizada para el cálculo del modelo de fondo actualizando el valor de la constante  $K$  en función del tamaño de la ventana, y porcentaje de píxeles segmentados de forma correcta.

Estos valores se muestran de forma gráfica en la Figura 5.33.

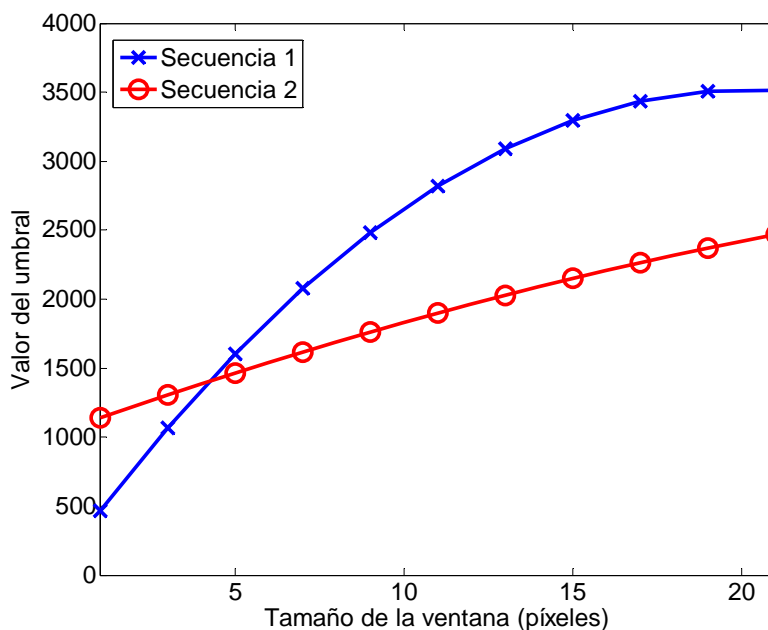


Figura 5.33. Valor inicial del umbral en función del tamaño de la ventana utilizada para el cálculo del modelo de fondo actualizando el valor de la constante  $K$  en función del tamaño de la ventana.



Si se compara la gráfica de la Figura 5.33 con la mostrada en la Figura 5.27 se puede observar como al ajustar el valor de la constante  $K$  en función del tamaño de la ventana utilizada en el cálculo del error de recuperación, el valor del umbral se incrementa de forma proporcional al tamaño de la ventana.

Si se representa el porcentaje de píxeles que se segmenta de forma correcta (considerando que la segmentación correcta es la realizada tras aplicar al error de recuperación de cada imagen el umbral mostrado en la Tabla 5.8) se obtienen las curvas mostradas en la Figura 5.34 en las que se puede observar que, incluso tras ajustar el valor de  $K$  en función del tamaño de la ventana, los resultados obtenidos para la imagen perteneciente a la secuencia 1 se deterioran rápidamente al incrementar el tamaño de ventana. En el caso de la secuencia 2, el ajuste de la constante  $K$  mejora los resultados respecto al caso de usar  $K=10$ .

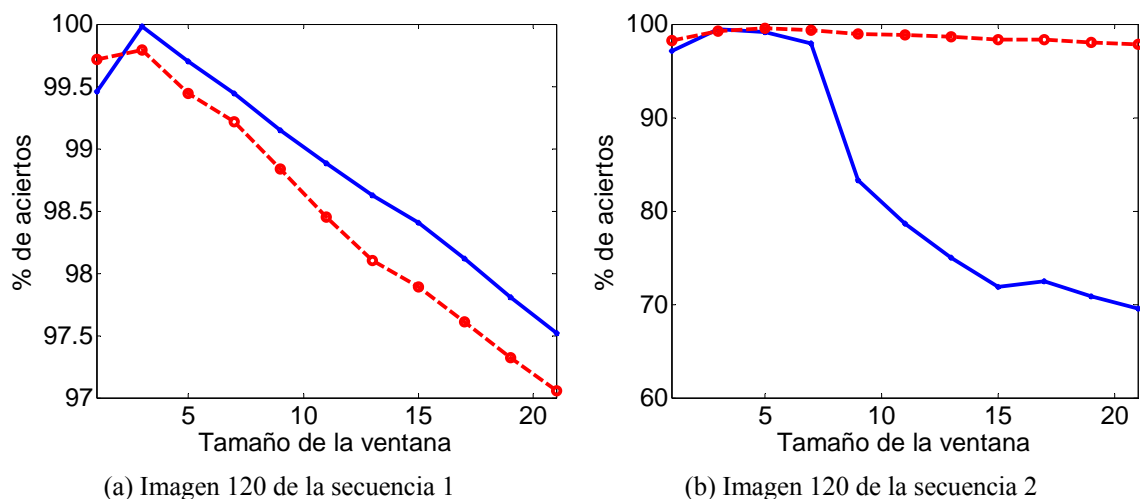


Figura 5.34. Porcentaje de píxeles segmentados de forma correcta en función del tamaño de la ventana utilizada en el cálculo del error de recuperación tras aplicar un umbral definido según la ecuación (5.14) con  $K$  constante (en línea azul continua) y con  $K$  definida en función del tamaño de ventana según las ecuaciones (5.15) y (5.16), respectivamente. (a) Imagen de la secuencia 1 (Figura 5.19 (a)). (b) Imagen de la secuencia 2 (Figura 5.19 (b))

Con respecto al tiempo de cómputo consumido en la obtención de las curvas iniciales, en la Figura 5.35 se muestra el valor medio de dicho tiempo calculado para 50 imágenes de cada una de las dos secuencias. Es evidente que, a medida que aumenta el tamaño de la ventana, el tiempo consumido en el cálculo y umbralización del error de recuperación es también mayor. El hardware utilizado corresponde al del nodo cliente, que ha sido definido en el apartado 3.1.1, al describir la arquitectura del ISPACE-UAH.

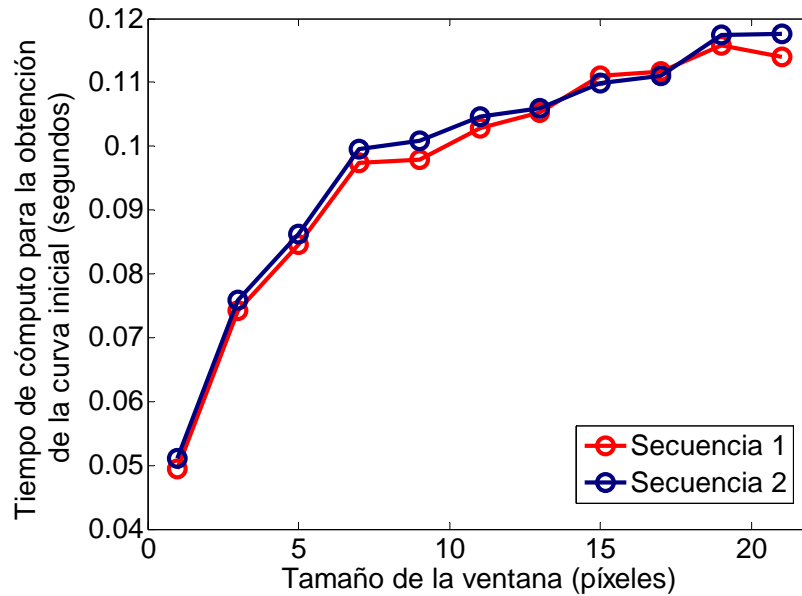


Figura 5.35. Tiempo de cómputo medio consumido en la obtención de la curva inicial en 50 imágenes en función del tamaño de la ventana utilizada en el cálculo del error de recuperación.

Las pruebas experimentales realizadas han puesto de manifiesto que, a pesar de haber ajustado el valor de la constante  $K$ , el incremento del tamaño de la ventana ( $q$ ) utilizada en el cálculo del error de recuperación no mejora la calidad de las curvas iniciales, e incluso la empeora en el caso de la secuencia 1 (Figura 5.34(a)). Además, el tiempo de procesamiento aumenta con el incremento del tamaño de ventana (Figura 5.35). Por los motivos expuestos se ha decidido utilizar una ventana de 3x3 píxeles, y mantener el valor de la constante de proporcionalidad del umbral (5.14),  $K=10$ .

### **Efecto de los cambios de iluminación sobre el umbral proporcional con $K$ constante**

Como ya se ha comentado, en este apartado se presentan los resultados de las pruebas experimentales realizadas para evaluar el efecto sobre las curvas iniciales de los cambios en la intensidad de las imágenes de entrada (producidos por cambios de la iluminación) en caso de utilizar el umbral proporcional con  $K$  constante (5.14).

Dado que los cambios en la iluminación se traducen en cambios de intensidad en la imagen en escala de grises, en este apartado se mostrará el efecto de los cambios en la intensidad media de las imágenes de entrada sobre las curvas iniciales, definiendo dichas curvas por el número de puntos que se encuentran en su interior.

Si se representa la intensidad media y el número de puntos que contiene la curva inicial (obtenida con el valor inicial del umbral) para 700 imágenes de cada una de las secuencias a las que pertenecen las imágenes de la Figura 5.19 (a) y (b), se obtienen las gráficas mostradas en la Figura 5.36 y la Figura 5.37. Tanto en la Figura 5.36 como en la Figura 5.37 se puede ver la relación entre el valor medio de la intensidad de la imagen de entrada en escala de grises y el número de puntos que contiene la curva inicial, para un valor fijo del umbral sobre el error de recuperación.

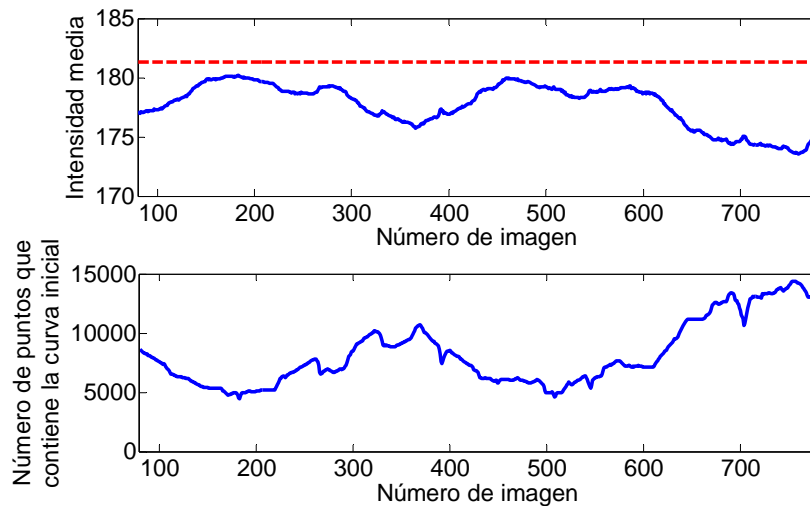


Figura 5.36. Intensidad media de las  $N_i$  imágenes utilizadas en el cálculo del modelo de fondo (en línea roja en la gráfica superior). Intensidad media y número de puntos contenidos por la curva inicial, aplicando el valor inicial del umbral proporcional (5.14) con  $K=10$  sobre el error de recuperación de 700 imágenes pertenecientes a la secuencia de la imagen mostrada en la Figura 5.11(a).

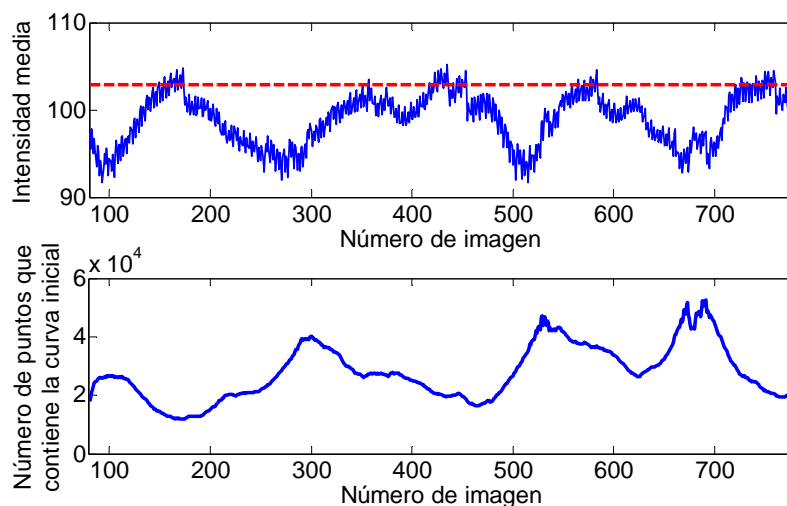


Figura 5.37. Intensidad media de las  $N_i$  imágenes utilizadas para el cálculo del modelo de fondo (en línea roja discontinua en la gráfica superior). Intensidad media y número de puntos contenidos por la curva inicial, aplicando el valor inicial del umbral proporcional (5.14) con  $K=10$  sobre el error de recuperación de 700 imágenes pertenecientes a la secuencia de la imagen mostrada en la Figura 5.11(b).

En ambas figuras se puede observar que a medida que se incrementa el valor de la intensidad media de la imagen de entrada, disminuye el número de puntos que contiene la curva inicial, es decir, la curva inicial se hace más pequeña, mientras que, si la intensidad media de la imagen de entrada disminuye, el número de puntos contenidos en la curva inicial aumenta.

Este efecto puede observarse también en la Figura 5.38, en la que se ha representado el número de puntos que contiene la curva inicial, en función de la intensidad media de la imagen de entrada para cada una de las imágenes de las dos secuencias. Es especialmente evidente en la Figura 5.38(a).

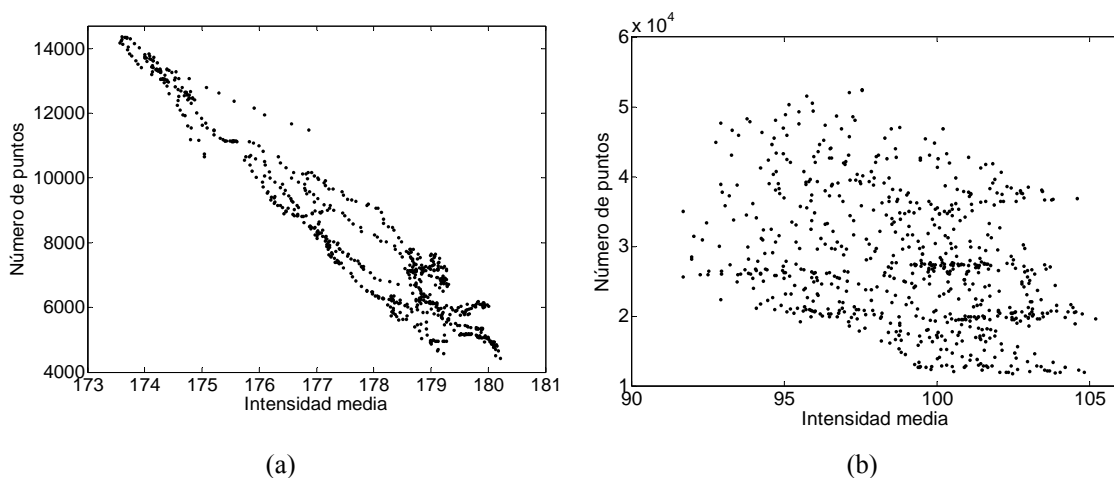


Figura 5.38. Número de puntos contenidos en la curva inicial en función de la intensidad media de la imagen de entrada. Estas gráficas se han obtenido representando para cada imagen el punto correspondiente a su intensidad media, y al número de píxeles en los que el error de recuperación supera el umbral fijado en la ecuación (5.14). En (a) se muestran los resultados para 700 imágenes de la secuencia 1 (Figura 5.11 (a)) mientras que en (b) se han representado los resultados para 700 imágenes de la secuencia 2 (Figura 5.11 (b))

Por otro lado, para evaluar el efecto de variaciones de la intensidad mayores que las mostradas en la Figura 5.36 y la Figura 5.37 sobre el error de recuperación, así como sobre el valor del umbral y las curvas iniciales, se ha modificado el valor de la intensidad media de las imágenes de entrada, incrementando (o decrementando) el valor de intensidad de cada uno de los píxeles de la imagen de entrada. En estas primeras pruebas, la variación en la intensidad se considera homogénea, de forma que el valor sumado (o restado) a todos los píxeles de la imagen es el mismo.

En la Figura 5.39(a) y (d) se muestran, a modo de ejemplo, dos imágenes del robot móvil pertenecientes a diferentes secuencias, así como las imágenes resultantes al restar (b) y (e) o sumar (c) y (f) un valor de 40 a la intensidad de cada uno de los píxeles de la imagen de entrada.

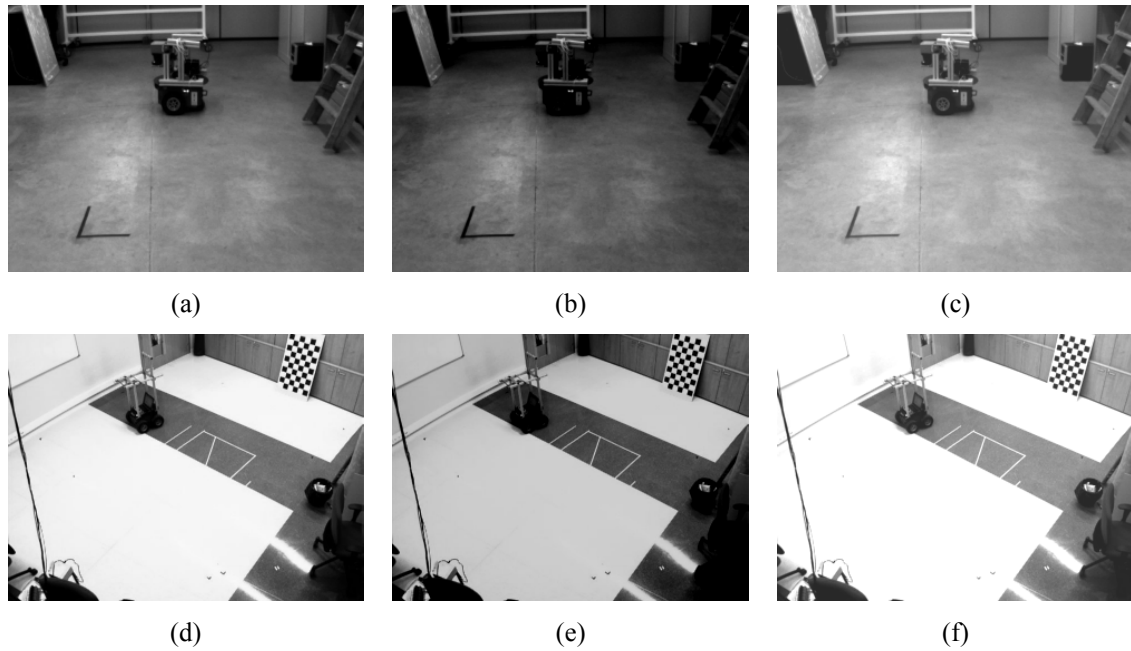


Figura 5.39. Ejemplo de las imágenes utilizadas para evaluar el efecto de los cambios de iluminación: (a) y (d) imágenes originales. (b) y (e) imágenes obtenidas tras restarle un valor de 40 a la intensidad de todos los píxeles de las imágenes originales. (c) y (f) imágenes obtenidas tras sumar un valor de 40 a la intensidad de todos los píxeles de las imágenes originales.

Con objeto de evaluar el efecto de los cambios en intensidad de las imágenes de entrada sobre las curvas iniciales, en la Figura 5.40 (a) se representa el número de píxeles que contiene la curva inicial, obtenida aplicando un umbral proporcional (5.14) al error de recuperación obtenido para una imagen del robot como la mostrada en la Figura 5.39 a la que se ha sumado un valor entre -200 y 200 a la intensidad de cada píxel (siendo el rango de niveles de gris de la imagen de 0 a 255).

En la Figura 5.40 (a) se puede observar como al sumar o restar valores pequeños (hasta  $\pm 25$ ) el número de píxeles se mantiene aproximadamente constante, debido a que se separa de forma correcta el robot móvil del fondo de la escena. Sin embargo, si se suma o resta un valor superior a 25, el número de píxeles contenidos en la curva inicial aumenta rápidamente, para luego mantenerse constante (zonas 1 y 3 en la Figura 5.40(a)). Esto se debe a que en caso de que se produzcan cambios grandes en la iluminación, el error de recuperación de los píxeles de fondo aumenta de forma significativa, de forma que algunos de estos píxeles del fondo superan el umbral fijado, pasando a considerarse como parte del robot móvil. Este mismo efecto se observa también en las Figura 5.40(b), (c) y (d) en las que se muestran las curvas iniciales en diferentes situaciones. En la Figura 5.40(b) Se presenta la curva inicial para la imagen original, mientras que en (a) y (c) se han representado las curvas iniciales para el caso

de sumar y restar, respectivamente, a la intensidad de los píxeles de la imagen de entrada un valor de 70.

A la vista de estos resultados se llega a la conclusión de que, en el caso de utilizar un valor fijo para la constante de proporcionalidad  $K$  del umbral (5.14) el margen de valores de intensidad válidos se reduce a la intensidad media de las imágenes del fondo  $\pm 25$ . Es decir, a la zona 2 de la Figura 5.40(a).

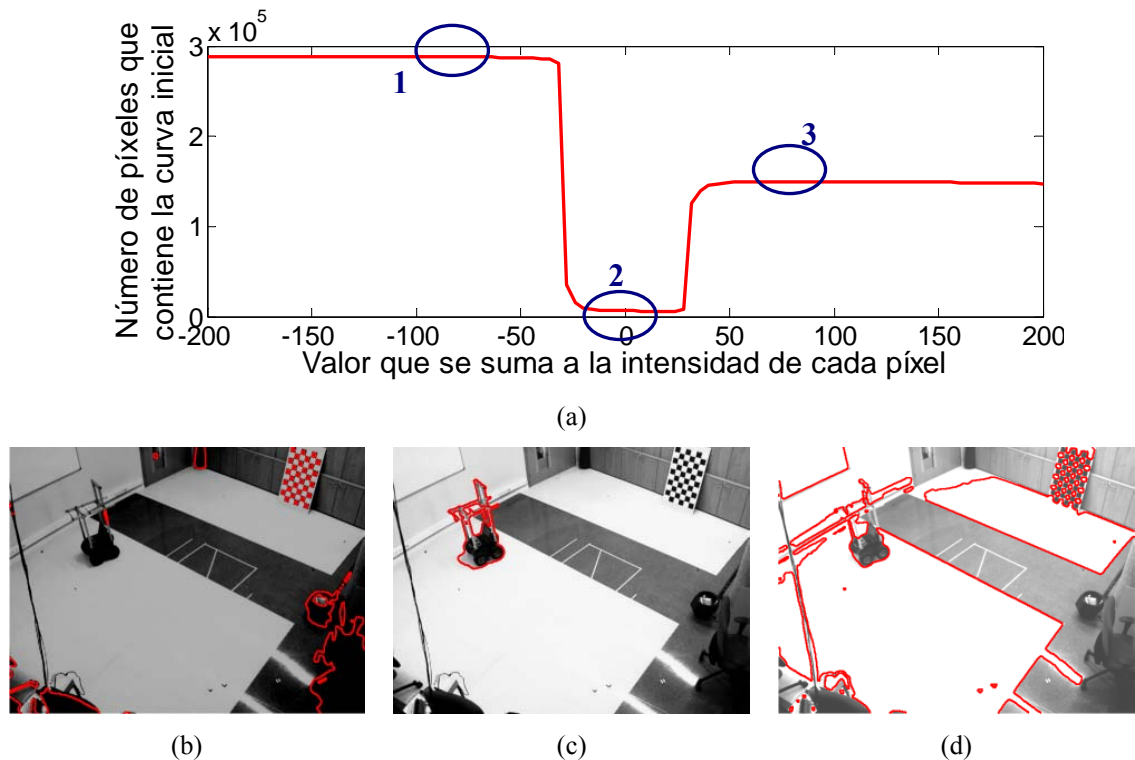


Figura 5.40. Efecto de la variación de la intensidad de las imágenes sobre las curvas iniciales. (a) Número de píxeles que contiene la curva inicial en función del valor que se suma a la intensidad en cada píxel de la imagen de entrada para un umbral proporcional con  $K = 10$ . (b) Curva inicial obtenida tras restar un valor igual a 70 a la intensidad de los píxeles de la imagen de entrada. (c) Curva inicial obtenida para la imagen de entrada original. (d) Curva inicial obtenida tras sumar un valor de 70 a la intensidad de los píxeles de la imagen de entrada.

En la Tabla 5.9 se presentan los resultados obtenidos al comparar las curvas iniciales obtenidas aplicando el umbral proporcional con  $K$  constante, definido en la ecuación (5.14), usando  $K=10$ , y las curvas iniciales obtenidas mediante el ajuste manual de los umbrales para los diferentes valores en los que se ha incrementado (o decrementando) la intensidad en los píxeles de las imágenes de prueba. En concreto, los porcentajes de acierto mostrados en la Tabla 5.9 se han obtenido como el valor medio de los resultados obtenidos para 10 imágenes de prueba. Como cabía esperar, a la vista de la curva de la Figura 5.40(a), en la Tabla 5.9 se observa de nuevo que si la intensidad de la imagen se incrementa en valores dentro del rango de  $\pm 25$ , el porcentaje de píxeles

segmentados correctamente es superior al 97% mientras que, fuera de ese rango de valores, el porcentaje de aciertos decrece rápidamente.

Incremento de I	Píxeles segmentados correctamente (%) $U_p$ (5.14) con $K=10$	Incremento de I	Píxeles segmentados correctamente (%) $U_p$ (5.14) con $K=10$
-10	98.2048 %	+10	98.4981 %
-20	97.9053 %	+20	98.6616 %
-30	97.3897 %	+30	98.7214 %
-40	96.5410 %	+40	96.2771 %
-50	93.7092 %	+50	87.4293 %
-60	88.0068 %	+60	80.7562 %
-70	81.1365 %	+70	75.7344 %
-80	75.0217 %	+80	71.3411 %
-90	69.5774 %	+90	67.4748 %
-100	64.5910 %	+100	63.9591 %

Tabla 5.9. Número de píxeles segmentados de forma correcta tras la umbralización del error de recuperación aplicando el umbral proporcional (5.14) con  $K=10$ , respecto al caso de la segmentación manual, cuando la secuencia de imágenes de entrada se ha modificado incrementando o reduciendo entre -100 y 100 el nivel de gris de todos los píxeles uniformemente.

Debido a que se pretende que el sistema sea robusto frente a cambios en la iluminación se propone utilizar un umbral adaptativo que tenga en cuenta estos cambios. Observando la Figura 5.36 y la Figura 5.37 se propone definir una función de ajuste del umbral, en función de la diferencia entre la intensidad media de la imagen de entrada y la intensidad media de las  $N_i$  imágenes empleadas en el cálculo del modelo de fondo. De esta forma, a medida que disminuya el valor medio de la intensidad de las imágenes de entrada, el valor del umbral debe aumentar con objeto de que el número de píxeles segmentados correctamente se mantenga aproximadamente constante.

### 5.2.2.2. Umbral adaptativo

En este apartado se propone actualizar el umbral en función de la diferencia entre el valor medio de intensidad de la imagen de entrada, y el valor medio de intensidad de las  $N_i$  imágenes utilizadas para calcular el modelo de fondo.

Bajo el supuesto de que la relación entre la diferencia de intensidades y el valor del umbral es lineal, y que a medida que la diferencia se incrementa (tanto si la intensidad media aumenta como si disminuye) el valor del umbral también debe incrementarse se define una nueva constante de proporcionalidad adaptativa ( $K_{ad}$ ), en función del valor

absoluto de la diferencia entre la intensidad media de la imagen de entrada, y la de las  $N_i$  imágenes de fondo data por la ecuación (5.17) en la que  $\alpha$  y  $\beta$  son dos constantes reales y positivas.

$$K_{ad} = \alpha + \beta |I - I_{media}^{fondo}| \quad \alpha, \beta \in \mathfrak{R}, \alpha \geq 0, \beta \geq 0 \quad (5.17)$$

Utilizando la expresión en (5.17) para la constante de proporcionalidad, el umbral adaptativo queda definido por la expresión de la ecuación (5.18).

$$U_{ad} = \left( \alpha + \beta |I - I_{media}^{fondo}| \right) \cdot \max \left( \left\{ \varepsilon_{r, fondo}(i, j) \right\}_{\substack{i=1, \dots, n_{filas} \\ j=1, \dots, n_{cols}}} \right) \quad \alpha, \beta \in \mathfrak{R}, \alpha, \beta \geq 0 \quad (5.18)$$

Mediante la ecuación (5.18) se pretende obtener un umbral que se incremente de forma proporcional al valor absoluto de la diferencia entre la intensidad media de la imagen de entrada, y la intensidad media de las  $N_i$  imágenes consideradas en el cálculo del modelo de fondo. Es decir, para la obtención del umbral se considerará únicamente la magnitud de la diferencia de intensidades, y no el signo de dicha diferencia. Esto es así porque, el valor del error de recuperación de los píxeles que pertenecen al fondo de la escena se incrementa de forma proporcional a la diferencia de intensidades, independientemente de si la intensidad de la imagen de entrada es mayor o menor que la de las imágenes de fondo.

### Efecto del valor de las constantes sobre las curvas iniciales

El umbral adaptativo (5.18) depende directamente de los valores de las constantes  $\alpha$  y  $\beta$ . A continuación se presentan los resultados obtenidos al estudiar el efecto de cada una de estas constantes sobre el valor del umbral y las curvas iniciales.

Para evaluar el efecto de la constante  $\alpha$  sobre las curvas iniciales se ha realizado un conjunto de pruebas experimentales en las que se ha sumado un valor entre -200 y 200 a todos los píxeles de la imagen de entrada. Se han obtenido las curvas iniciales en esas imágenes, aplicando un umbral dinámico dado por (5.17) obtenido para diferentes valores de  $\alpha$  manteniendo fijo el valor de  $\beta$  en 0.75. En la Figura 5.41 se representa el número de píxeles contenidos en la curva inicial que se ha obtenido tanto aplicando un umbral fijo, como empleando el umbral dinámico en función de la intensidad para  $\alpha = [1, 2.5, 5, 7.5, 9]$ . En Figura 5.41 se puede observar como el rango de valores en los que se obtiene un resultado adecuado (no se incorporan píxeles del fondo a las curvas



iniciales) aumenta a medida que se incrementa el valor de la constante  $\alpha$  manteniendo fijo el valor de  $\beta$ .

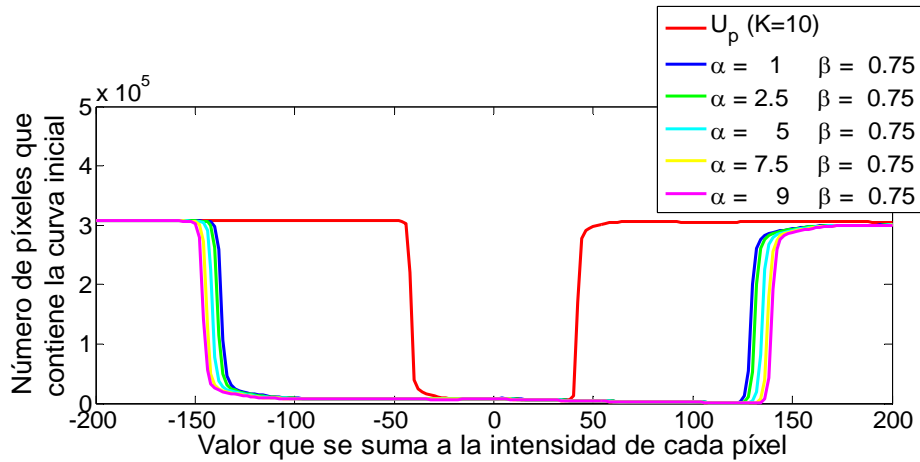


Figura 5.41. Efecto de la constante  $\alpha$  sobre las curvas iniciales: número de píxeles contenidos en la curva inicial en función del valor sumado a la intensidad de todos los píxeles de la imagen de entrada para diferentes valores de la constante  $\alpha$  con  $\beta = 0.75$ .

Por otro lado, si se representa, de nuevo, el número de píxeles incluidos en la curva inicial en función de la constante  $\beta$ , manteniendo fijo el valor de  $\alpha$ , se obtiene la gráfica mostrada en la Figura 5.42, donde se puede observar como el aumento del valor de  $\beta$  también incrementa el rango de valores de intensidad en los que es posible discriminar entre los píxeles que pertenecen al robot, y aquellos que corresponden al fondo de la escena.

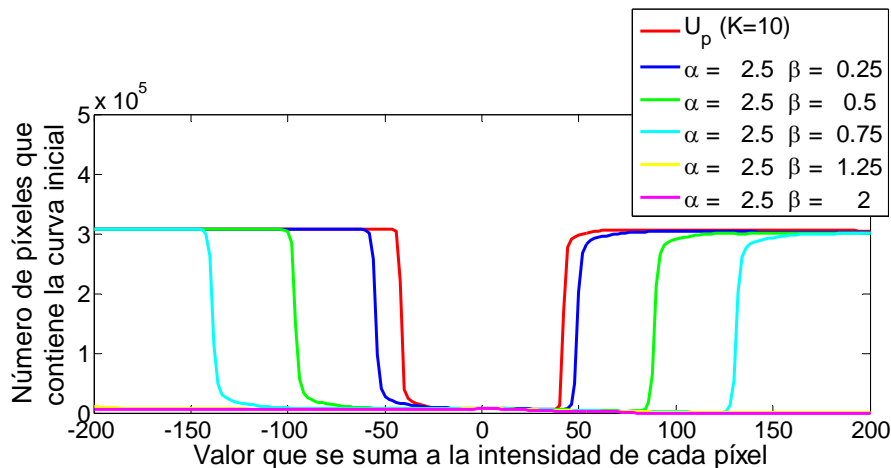


Figura 5.42. Efecto de la constante  $\beta$  sobre las curvas iniciales: número de píxeles contenidos en la curva inicial en función del valor sumado a la intensidad de todos los píxeles de la imagen de entrada para diferentes valores de la constante  $\beta$  con  $\alpha = 2.5$ .

Tanto en la Figura 5.41 como en la Figura 5.42 se puede observar que el umbral adaptativo propuesto proporciona resultados adecuados frente a cambios en la iluminación. Este umbral es válido tanto si se disminuye, como si se aumenta el valor de

la intensidad de forma aproximadamente simétrica con respecto al valor 0 que se corresponde con la imagen de entrada original.

Con objeto de evaluar un conjunto mayor de valores de las constantes  $\alpha$  y  $\beta$ , a continuación, en la Figura 5.43 se representa el rango de valores en los que las curvas iniciales se obtienen de forma correcta. En este caso, debido a que las curvas son aproximadamente simétricas (según se ha podido observar en la Figura 5.42) en lugar de mostrar los valores positivos y negativos, se ha representado el rango de valores. Este rango se define como la diferencia entre el valor máximo que puede sumarse a la intensidad y el valor máximo que puede restarse a la misma, sin que las curvas iniciales se degraden.

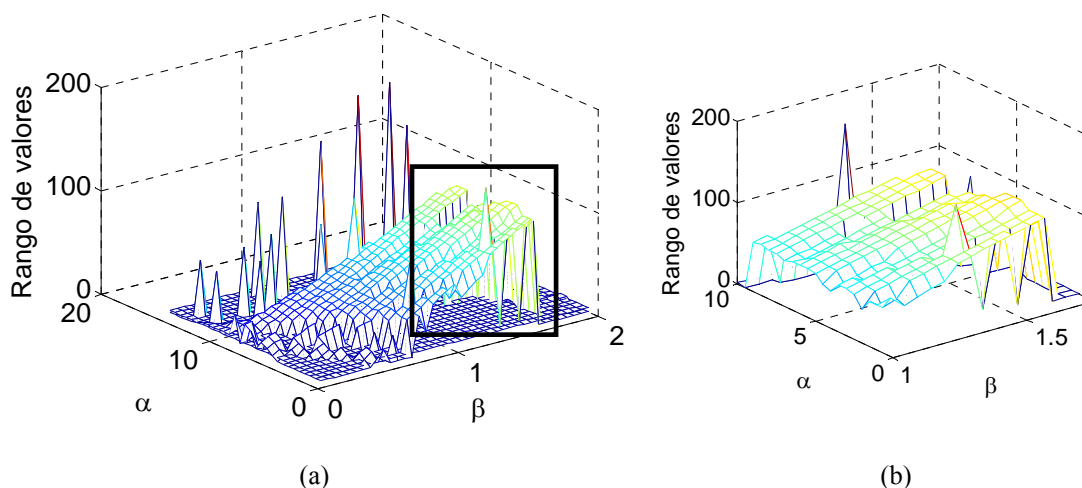


Figura 5.43. (a) Rango de valores en los que es posible incrementar o decrementar la intensidad en cada píxel de la imagen de entrada obteniendo las curvas iniciales de forma correcta en función de los valores de las constantes  $\alpha$  y  $\beta$ . (b) Detalle del rango de valores para  $\alpha$  entre 1 y 11 y  $\beta$  entre 1 y 1.8.

A continuación, en la Tabla 5.10 se muestran los valores de algunos puntos de la gráfica mostrada en la Figura 5.43(b). Los valores presentados aquí se han obtenido como la media de los resultados obtenidos para 9 imágenes de entrada, concretamente las imágenes 100, 150, 200, 250, 300, 350, 400, 450 y 500. El rango de valores se calcula como la diferencia entre el máximo valor en el que se puede incrementar la intensidad de cada píxel de la imagen de entrada, y el valor máximo en que se puede decrementar dicha intensidad, sin que la segmentación se deteriore de forma significativa. En concreto, se considera un valor de intensidad dentro del rango válido únicamente si la curva inicial contiene un número de píxeles que no difiere en más de un 5% del número de píxeles obtenidos en la segmentación manual y además, el porcentaje de píxeles segmentados correctamente debe ser superior al 98%.

	$\alpha=0.6$	$\alpha=0.8$	$\alpha = 1$	$\alpha = 3$	$\alpha = 5$	$\alpha = 7$	$\alpha = 9$	$\alpha = 11$
$\beta = 1.0$	70.6	66.6	70.6	59.4	52	39.4	32.4	32.2
$\beta = 1.2$	72.4	72	85	55	45.2	36	35.2	35.6
$\beta = 1.4$	75.2	69.4	53.8	42.4	43	41.2	40.2	39.6
$\beta = 1.6$	72.8	72.8	43.2	41.8	35.4	33.8	33.4	32.6
$\beta = 1.8$	81.8	81.2	38.4	36.2	35.2	29	28.6	27.8

Tabla 5.10. Magnitud del rango de valores en que se puede incrementar la intensidad de las imágenes de entrada sin que se deterioren las curvas iniciales obtenidas en función del valor de las constantes  $\alpha$  y  $\beta$  consideradas en el cálculo del umbral (5.18).

A la vista de los resultados anteriores, si únicamente se tiene en cuenta el rango de valores de intensidad para los que se obtiene una segmentación adecuada de las imágenes, se llega a la conclusión de que los valores más adecuados de las constantes  $\alpha$  y  $\beta$  para el cálculo del valor del umbral son los que aparecen en negrita en la Tabla 5.10. En concreto se ha elegido el valor de la constante  $\alpha$  entre 0.6 y 1 y el de  $\beta$  entre 1 y 1.4.

A modo de ejemplo, en la Figura 5.44 se muestran las curvas iniciales obtenidas al aplicar el umbral dinámico sobre el error de recuperación para diferentes valores de las constantes.

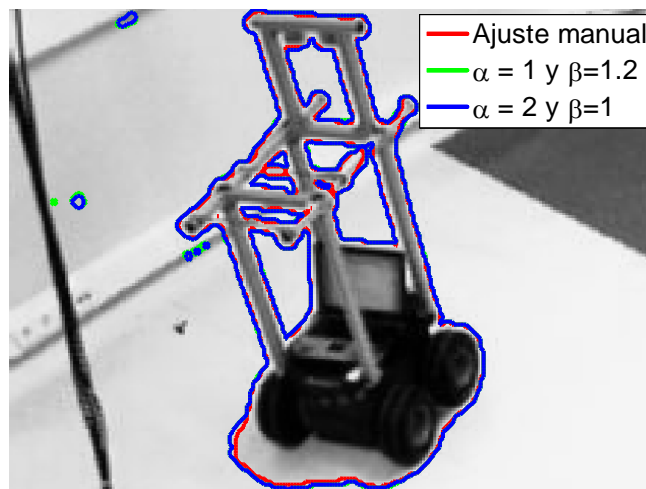


Figura 5.44. Detalle de las curvas iniciales obtenidas tras aplicar sobre el error de recuperación el umbral dinámico obtenido para diferentes valores de las constantes  $\alpha$  y  $\beta$ .

Por todo esto se ha decidido utilizar un umbral adaptativo definido según la ecuación (5.18) en la que los valores de las constantes se establecen en  $\alpha = 2$  y  $\beta = 1$ . A continuación se presentan los resultados de los experimentos realizados con objeto de determinar la bondad de los valores elegidos para las constantes  $\alpha$  y  $\beta$ .

En primer lugar, en la Tabla 5.11 se muestra el porcentaje de píxeles de la imagen en los que la curva inicial obtenida tras ajustar el valor del umbral en cada imagen de

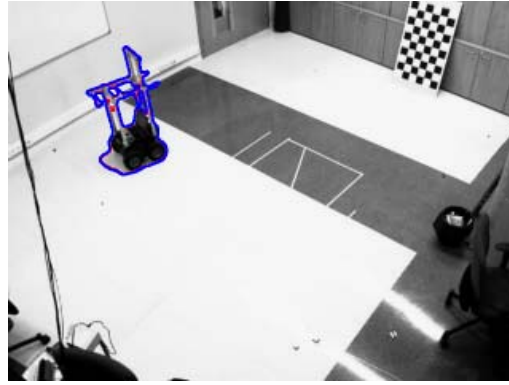
forma manual coincide con la curva inicial calculada con el umbral adaptativo. El valor presentado en la tabla es la media del obtenido para 10 imágenes de una secuencia de entrada, concretamente las imágenes 100, 125, 150, 175, 200, 225, 250, 275 y 300. La segmentación obtenida utilizando el umbral dinámico propuesto según la ecuación (5.18) con el valor de  $K$  definido por la función (5.17) se compara con el resultado obtenido aplicando un umbral proporcional (5.14) con  $K = 10$ .

En la Tabla 5.11 se puede observar que, a pesar de que el cambio de la intensidad es importante (la intensidad de cada píxel se modifica hasta en un valor de 50) al utilizar el umbral adaptativo ( $U_{ad}$ ) (5.18) se obtienen buenos resultados. Mientras que, en el caso de aplicar un umbral proporcional ( $U_p$ ) (5.14) con  $K=10$ , los resultados de la inicialización de las curvas se degradan rápidamente con las variaciones en la intensidad media de las imágenes de entrada.

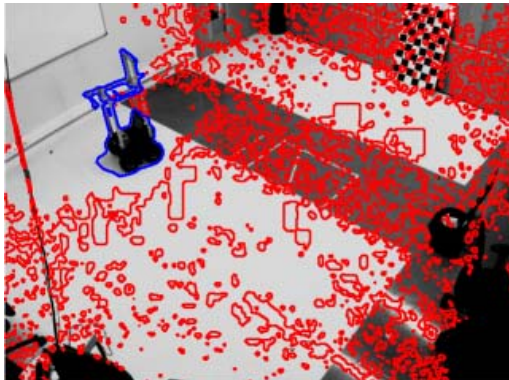
Valor sumado a la intensidad	% Píxeles segmentados correctamente		Valor sumado a la intensidad	% Píxeles segmentados correctamente	
	$U_p$	$U_{ad}$		$U_p$	$U_{ad}$
-50	0.0891 %	0.9961 %	50	0.5355 %	0.9630 %
-40	0.0920 %	0.9963 %	40	0.5468 %	0.9899 %
-30	0.6269 %	0.9963 %	30	0.8233 %	0.9921 %
-20	0.9920 %	0.9964 %	20	0.9951 %	0.9939 %
-10	0.9974 %	0.9968 %	10	0.9961 %	0.9963 %
0	0.9970 %	0.9981 %			

Tabla 5.11. Número de píxeles (en porcentaje) segmentados de forma correcta tras incrementar la intensidad de todos los píxeles de la imagen de entrada en un valor entre -50 y +50.

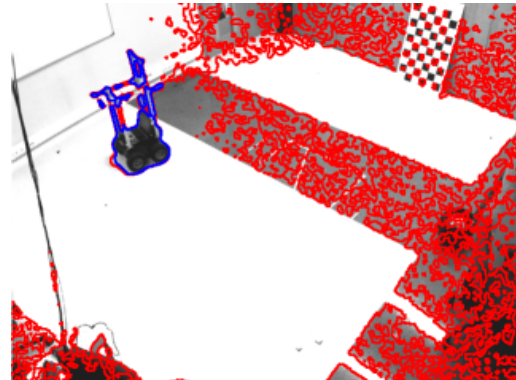
Por otro lado, en la Figura 5.45 se muestran las curvas iniciales en diferentes situaciones tanto para el umbral proporcional (en rojo) como para el umbral adaptativo propuesto (en azul). Concretamente se han mostrado las curvas obtenidas al umbralizar el error de recuperación calculado a partir de la imagen original tras sumarle un valor de: -50, -30, 30 y 50 a la intensidad de todos los píxeles. En todos los casos se puede observar cómo el uso del umbral adaptativo mejora de forma significativa los resultados obtenidos permitiendo diferenciar correctamente el robot del fondo de la escena en todos los casos presentados.



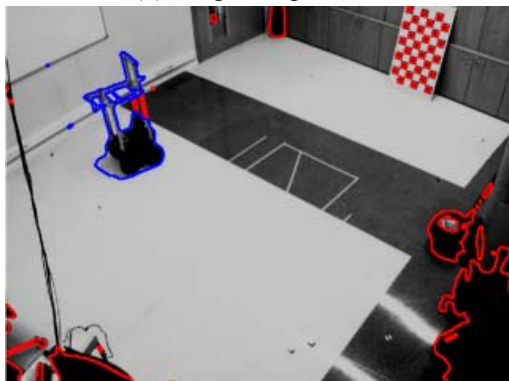
(a) Imagen original



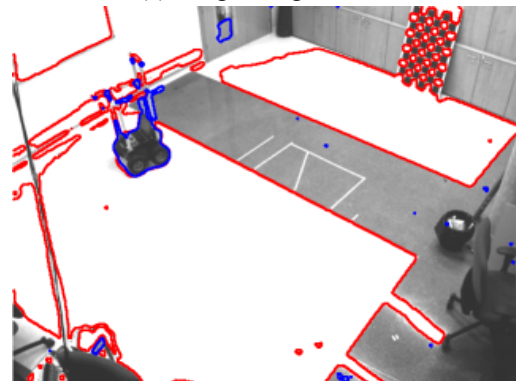
(b) Imagen original -30



(c) Imagen original +30



(d) Imagen original -50



(e) Imagen original +50

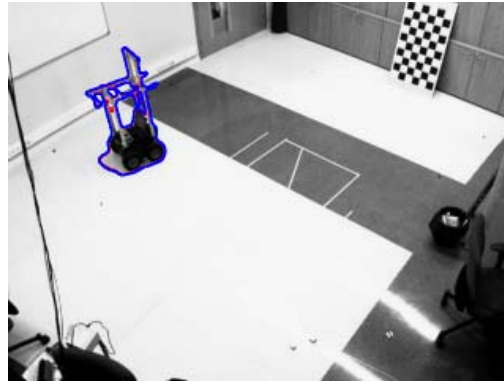
Figura 5.45. Curvas iniciales obtenidas aplicando el umbral proporcional (5.14) con  $K=10$  y el umbral adaptativo (5.18) al error de recuperación obtenido en imágenes con diferentes valores de intensidad media para una variación homogénea de la intensidad. En rojo se muestra el resultado obtenido al utilizar un umbral fijo (ecuación (5.14)) mientras que en azul se muestran las curvas iniciales obtenidas con el umbral dinámico (ecuación (5.18)).

Además, con objeto de evaluar la validez del umbral adaptativo propuesto, se han realizado diversas pruebas en las que se ha modificado la intensidad de las imágenes de entrada de forma no homogénea. Es decir, se ha sumado o restado a la intensidad de cada píxel de la imagen un valor diferente comprendido entre cero, y un valor máximo (positivo o negativo). Los resultados obtenidos se presentan a continuación. Al igual que en el caso anterior se ha representado la intensidad media de las imágenes de

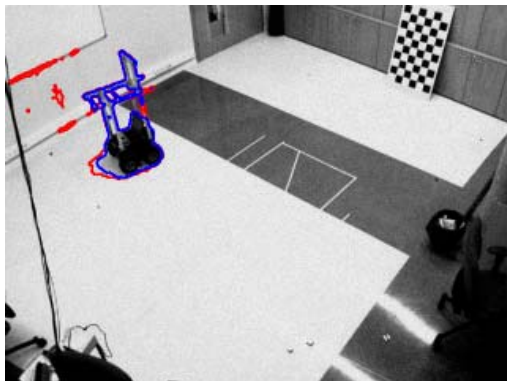


entrada, el porcentaje de píxeles contenidos en la curva inicial, así como las curvas iniciales para algunos casos.

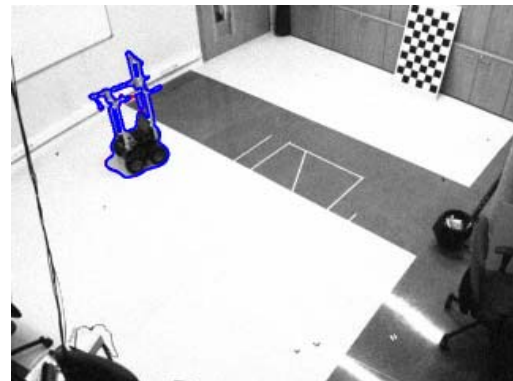
En la Figura 5.46 se muestran las curvas iniciales obtenidas para la imagen original (Figura 5.46(a)) así como para 4 imágenes que se han obtenido tras sumar (o restar) a cada píxel un valor comprendido entre 0 y -30 ( Figura 5.46(b)), 0 y +30 (Figura 5.46(c)), 0 y -50 (Figura 5.46(d)), 0 y +50 (Figura 5.60.e).



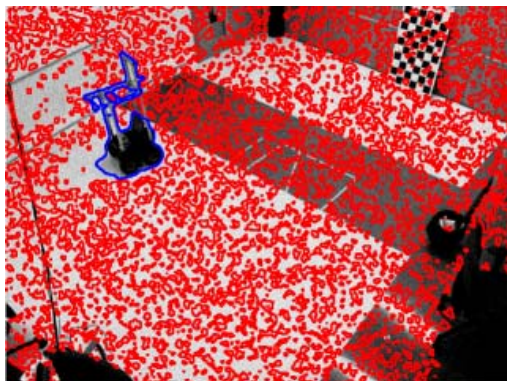
(a) Imagen original



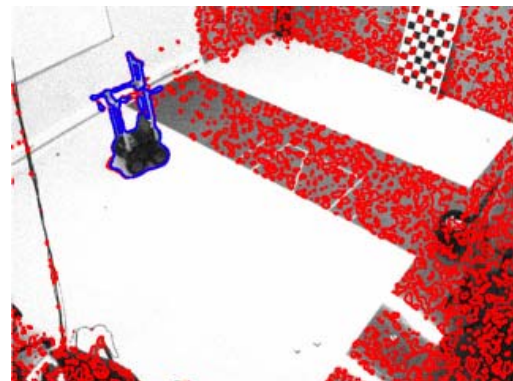
(b) Imagen original -30



(c) Imagen original +30



(d) Imagen original -50



(e) Imagen original +50

Figura 5.46. Curvas iniciales obtenidas aplicando el umbral proporcional (5.14) con  $K=10$  (en rojo) y el umbral adaptativo (5.18) (en azul) al error de recuperación obtenido en imágenes a las que se le ha modificado la intensidad de forma no homogénea.

En todas las imágenes mostradas en la Figura 5.46 se puede comprobar que a pesar de que la intensidad de la imagen de entrada no se haya modificado de forma homogénea, el umbral dinámico propuesto proporciona mejores resultados que el umbral fijo propuesto inicialmente. Esto es especialmente evidente en la Figura 5.46 (d) y (e), en las que se puede observar cómo el umbral proporcional con  $K$  constante no permite discriminar entre los píxeles del robot móvil y aquellos que pertenecen al fondo.

El mismo efecto puede observarse en la Tabla 5.12 en la que se muestra el porcentaje de píxeles segmentados de forma correcta al sumar a cada píxel de una imagen de entrada un valor aleatorio entre 0 y un valor máximo dado utilizando 50 utilizando el umbral proporcional con  $K$  constante (5.14) y el umbral adaptativo propuesto (5.18).

Valor máximo sumado a la intensidad	% Píxeles segmentados correctamente		Valor máximo sumado a la intensidad	% Píxeles segmentados correctamente	
	$U_p$	$U_{ad}$		$U_p$	$U_{ad}$
-50	0.6396	0.9964	50	0.8595	0.9916
-40	0.9692	0.9965	40	0.9891	0.9943
-30	0.9952	0.9967	30	0.9958	0.9953
-20	0.9973	0.9969	20	0.9962	0.9966
-10	0.9972	0.9974	10	0.9966	0.9977
0	0.9970	0.9981			

Tabla 5.12. Número de píxeles (en porcentaje) segmentados de forma correcta tras incrementar la intensidad de cada píxel de la imagen de entrada en un valor aleatorio entre -50 y +50 utilizando un umbral proporcional con  $K$  constante (5.14) y un umbral adaptativo (5.18).

### Validación del umbral adaptativo

En la Figura 5.47 se representan las curvas iniciales para una imagen de la secuencia 2 (Figura 5.19 (b)). En la Figura 5.47 (a) se muestra la inicialización con una circunferencia (como se propone en [Sekkati & Mitiche 2006b]) mientras que a la derecha se presenta la curva obtenida mediante GPCA. Se puede observar como la curva inicial utilizando la técnica propuesta en esta tesis se aproxima al contorno del robot. Como se ha visto a lo largo de este apartado, la curva inicial depende fuertemente del valor del umbral utilizado, sin embargo, en cualquier caso, el contorno obtenido mediante GPCA siempre se aproxima más al contorno real que una circunferencia

(excepto si el objeto es circular). El hecho de que las curvas iniciales se aproximen al contorno real del robot se traduce en una reducción del número de iteraciones necesarias hasta la convergencia del algoritmo de minimización de energía, reduciéndose también de forma significativa el tiempo de cómputo necesario para la segmentación de movimiento de los robots.



(a) Circunferencia

(b) Curva obtenida usando GPCA

Figura 5.47. Curvas iniciales obtenidas (a) utilizando una circunferencia según se propone en [Sekkati & Mitiche 2006b], y (b) utilizando la técnica propuesta en esta tesis basada en la umbralización del error de recuperación calculado usando la técnica GPCA.

En la Figura 5.47 (b) puede apreciarse que la sombra proyectada por el robot se ha considerado como parte del mismo, incluyéndose en la curva inicial. La inclusión de esta sombra puede introducir errores importantes a la hora de estimar la posición 3D de los objetos móviles presentes en la escena mediante el algoritmo propuesto en esta tesis. Con objeto de minimizar el efecto de las sombras en la segmentación y el posicionamiento 3D, se ha incluido en la etapa de inicialización de las curvas un proceso de eliminación de las sombras de forma que estas no se incluyan dentro de las curvas que se utilizarán en las siguientes etapas del algoritmo de segmentación propuesto.

Para validar el umbral propuesto en este apartado (definido en (5.18)) se han realizado también pruebas experimentales con imágenes pertenecientes a otras secuencias. Algunas de las imágenes con las que se han realizado los experimentos se muestran en la Figura 5.48. Estas imágenes pertenecen a dos secuencias diferentes capturadas en el espacio inteligente.



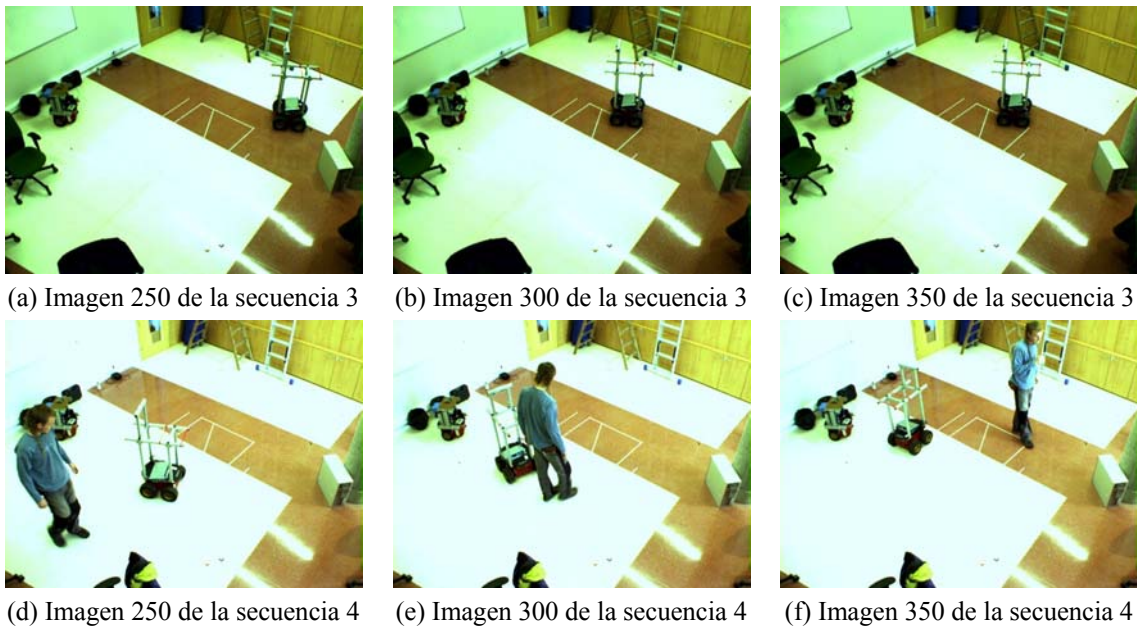


Figura 5.48. Imágenes de prueba pertenecientes a las secuencias 3 y 4. (a) Imagen 250 de la secuencia 3. (b) Imagen 300 de la secuencia 3. (c) Imagen 350 de la secuencia 3. (d) Imagen 250 de la secuencia 4. (e) Imagen 300 de la secuencia 4. (f) Imagen 350 de la secuencia 4.

A continuación se muestran las curvas iniciales obtenidas tras aplicar el umbral propuesto en la ecuación (5.18) a las imágenes mostradas en la Figura 5.48. Todas las curvas mostradas se han obtenido utilizando los mismos valores de  $K$ ,  $\alpha$  y  $\beta$  que han sido fijados para las imágenes de la secuencia 1. Se ha decidido utilizar los mismos valores debido a que todas estas secuencias de imágenes (1, 3 y 4) han sido adquiridas en el espacio inteligente, empleando para ello las mismas cámaras calibradas.

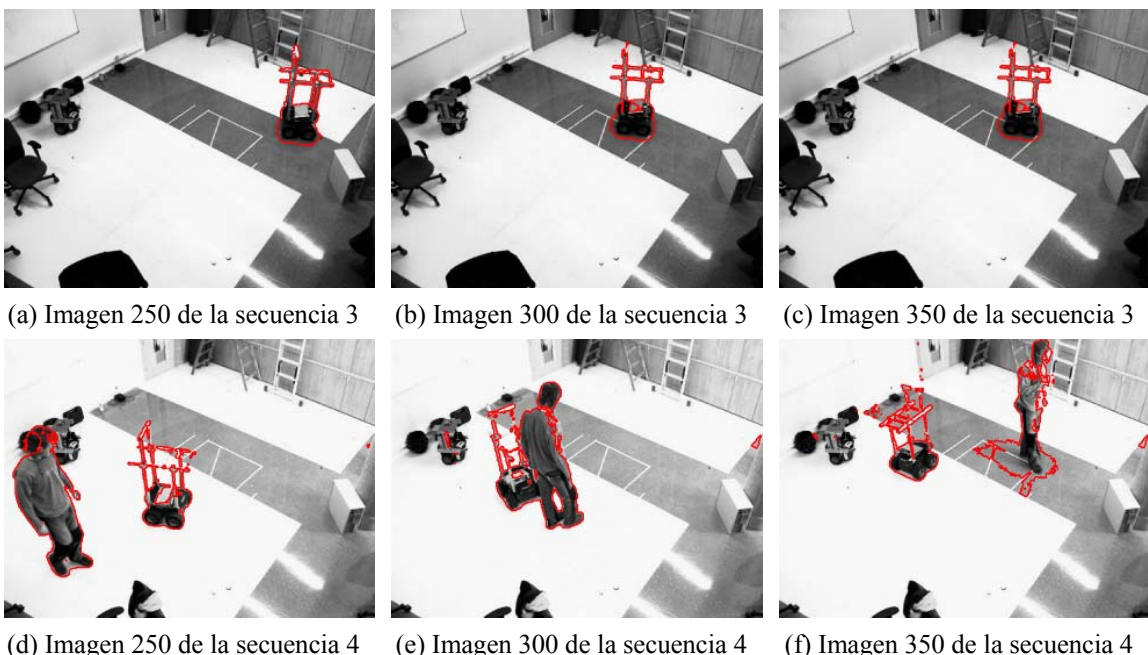


Figura 5.49. Curvas iniciales obtenidas para las imágenes de prueba de las secuencias 3 y 4 mostradas en la Figura 5.48. (a) Imagen 250 de la secuencia 3. (b) Imagen 300 de la secuencia 3. (c) Imagen 350 de la secuencia 3. (d) Imagen 250 de la secuencia 4. (e) Imagen 300 de la secuencia 4. (f) Imagen 350 de la secuencia 4.

Finalmente, se presenta también el porcentaje de píxeles inicializados de forma correcta para las imágenes consideradas en cada una de las dos secuencias. Para determinar este porcentaje se ha realizado la segmentación manual de algunas imágenes (según se explicó anteriormente para las imágenes de las secuencias 1 y 2) y se han comparado esas imágenes umbralizadas manualmente, con las obtenidas mediante el umbral adaptativo propuesto (5.18).

En la Tabla 5.9 se muestra el porcentaje de píxeles segmentados de forma correcta, calculado como el valor medio de los porcentajes para las tres imágenes de las secuencias 3 y 4 mostradas en la Figura 5.48.

<b>Incremento de I</b>	<b>Píxeles segmentados correctamente (%). Secuencia 3 (Figura 5.48 (a))</b>	<b>Píxeles segmentados correctamente (%). Secuencia 4 (Figura 5.48 (d))</b>
-50	99.6692	98.6324
-40	99.6536	99.1368
-30	99.6449	99.3108
-20	99.6542	99.4133
-10	99.6858	99.5029
0	99.6763	99.4225
10	99.6293	99.4617
20	99.3477	99.1795
30	99.0903	98.7173
40	98.7174	98.2862
50	98.3256	97.8189

Tabla 5.13. Número de píxeles segmentados de forma correcta tras la umbralización del error de recuperación aplicando el umbral adaptativo definido en la ecuación (5.18) sobre las imágenes de entrada pertenecientes a las secuencias 3 y 4 (mostradas en la Figura 5.48) a las que se les ha modificado la intensidad media sumando un valor aleatorio entre -50 y 50 a cada píxel de cada imagen.

A continuación, en la Figura 5.50 se muestra gráficamente el porcentaje de píxeles segmentados de forma correcta para las imágenes de las secuencias mostradas en la Figura 5.48, tras aplicar un umbral adaptativo (5.18) sobre el error de recuperación. Las imágenes han sido modificadas sumando o restando a todos los píxeles de la imagen un valor dentro del rango  $\pm 150$ . No se ha considerado necesario representar el caso de sumar un valor aleatorio a cada píxel debido a que se ha comprobado que, en este caso, los resultados son mejores que si se modifica la intensidad de toda la imagen por igual.

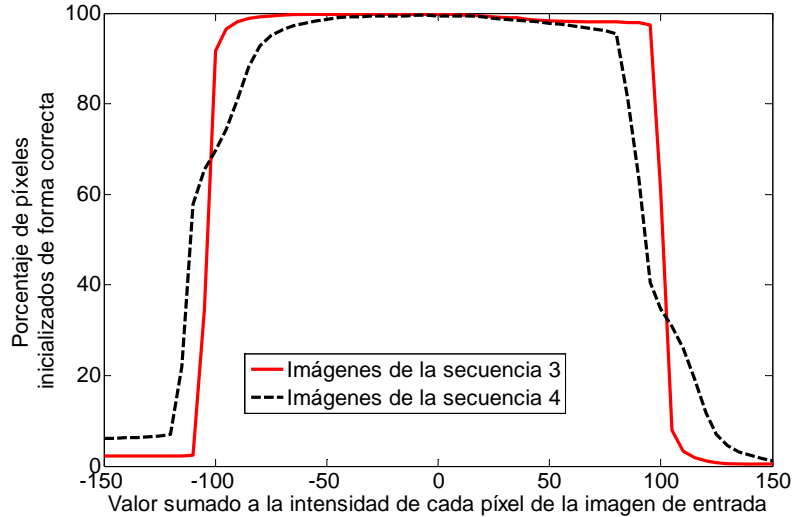


Figura 5.50. Porcentaje de píxeles segmentados de forma correcta en función del valor sumado a la intensidad de cada píxel de la imagen de entrada. Resultados obtenidos para las imágenes de la secuencia 3 (en rojo en la figura) y la secuencia 4 (en negro en la figura).

### 5.3. Eliminación de sombras

Como ya se ha comentado, la presencia de sombras en la imagen introduce errores importantes en el posicionamiento 3D de los objetos. Para tratar de minimizar este efecto, se introduce una nueva etapa en la inicialización de las curvas cuyo objetivo es la detección y eliminación de las sombras en las curvas iniciales. Para la detección de las sombras se emplea una representación de la imagen en escala de grises invariante a la iluminación [Finlayson et al. 2002], [Finlayson et al. 2006].

Para la obtención de la imagen invariante se supone un modelo *Lambertiano* de formación de la imagen de forma que, si denominamos  $E(\lambda, x, y)$  a la distribución espectral de potencia que incide sobre una superficie cuya función de reflectancia es  $S(\lambda, x, y)$ , la respuesta de los sensores de la cámara puede expresarse como:

$$R_k(x, y) = \sigma(x, y) \int E(\lambda, x, y) S(\lambda, x, y) Q_k(\lambda) d\lambda \quad (5.19)$$

donde  $Q_k(\lambda)$  es la sensibilidad espectral del sensor  $k$  ( $k = 1, 2, 3$ ) y  $\sigma(x, y)$  es un factor constante obtenido como el producto escalar del vector normal a la superficie y la dirección de la iluminación.

La imagen invariante a la iluminación se obtiene bajo dos supuestos. En primer lugar la respuesta de cada uno de los sensores de la cámara se considera una delta de Dirac:  $Q_k = q_k \delta(\lambda - \lambda_k)$ , siendo  $q_k = Q_k(\lambda_k)$ . Bajo esta suposición, la ecuación (5.19)

puede reducirse a (5.20). Por otro lado, también se supone que la iluminación puede aproximarse por la ley de Planck (5.21), donde las constantes  $c_1 = 3.74183 \times 10^{-6} \text{ Wm}^2$  y  $c_2 = 1.4388 \times 10^{-2} \text{ mK}$ , mientras que  $I$  depende de la intensidad de luz incidente.

$$R_k = \sigma E(\lambda_k) S(\lambda_k) q_k \quad (5.20)$$

$$E(\lambda, T) = I c_1 \lambda^{-5} \left( e^{\frac{c_2}{T\lambda}} - 1 \right)^{-1} \quad (5.21)$$

Para sistemas de iluminación en el rango de temperatura de 2500K a 10000K se cumple que  $e^{\frac{c_2}{T\lambda}} \gg 1$  de forma que puede utilizarse la aproximación de Wien (5.22):

$$E(\lambda, T) \approx I c_1 \lambda^{-5} e^{-\frac{c_2}{T\lambda}} \quad (5.22)$$

Sustituyendo en la ecuación (5.20)  $E(\lambda, T)$  por la expresión aproximada (5.22) se obtiene la siguiente expresión para la respuesta de cada uno de los canales de color:

$$R_k = I c_1 \lambda^{-5} e^{-\frac{c_2}{T\lambda}} S(\lambda_k) q_k \quad (5.23)$$

El modo más sencillo de eliminar la dependencia de la iluminación es dividir dos de los canales de color entre el tercero de ellos, obteniendo así un espacio de dimensión 2. Sin embargo, esta división puede sesgar los resultados favoreciendo a una de las componentes de color frente a las demás. Para evitar este efecto se divide la respuesta de cada uno de los canales de color entre la media geométrica de los tres según se indica en [Finlayson et al. 2004], obteniéndose la ecuación (5.24) para cada canal de color, y su versión logarítmica (5.25).

$$c_k = \frac{R_k}{\sqrt[3]{\prod_{i=1}^3 R_i}} = \frac{R_k}{R_M} \quad k = 1, 2, 3 \quad (5.24)$$

$$\rho_k = \log(c_k) = \log\left(\frac{s_k}{s_M}\right) + \frac{1}{T}(e_k - e_M) \quad k = 1, 2, 3 \quad (5.25)$$

siendo:

$$s_k = k_1 \lambda_k^{-5} S(\lambda_k) q_k$$

$$e_k = -k_2 / \lambda_k$$

$$s_M = \sqrt{\prod_{j=1}^3 s_j}$$

$$e_M = -k_2 / \sqrt[3]{\sum_{j=1}^3 \lambda_j}$$

Dado que el espacio logarítmico  $\boldsymbol{\rho} = (\rho_R, \rho_G, \rho_B)^T$  define un plano en  $\mathfrak{R}^3$  (Figura 5.51) la matriz de covarianza de  $\boldsymbol{\rho}$  tiene únicamente dos autovalores no nulos, cuyos autovectores asociados son las columnas que forman la matriz ortogonal  $\mathbf{U}$ , de dimensión  $2 \times 3$ , que permite realizar la transformación del espacio de color 3D a un espacio 2D usando la expresión (5.26):

$$\boldsymbol{\chi} = \mathbf{U} \cdot \boldsymbol{\rho} \quad (5.26)$$

Por otro lado, en la ecuación (5.25), el primer término depende tanto de la superficie como de la cámara, pero es independiente de la iluminación, mientras que el segundo es independiente de la superficie, pero no de la cámara. A la vista de esta ecuación, se puede observar que ante cambios en el color de la iluminación (variaciones de  $T$ ) el vector  $\boldsymbol{\rho} = (\rho_R, \rho_G, \rho_B)^T$  para una superficie dada se desplaza a lo largo de una línea recta cuya dirección depende únicamente de las características de la cámara. Dichas líneas se encuentran sobre el plano definido por  $\boldsymbol{\rho}$  en  $\mathfrak{R}^3$ , mostrado en la Figura 5.51.

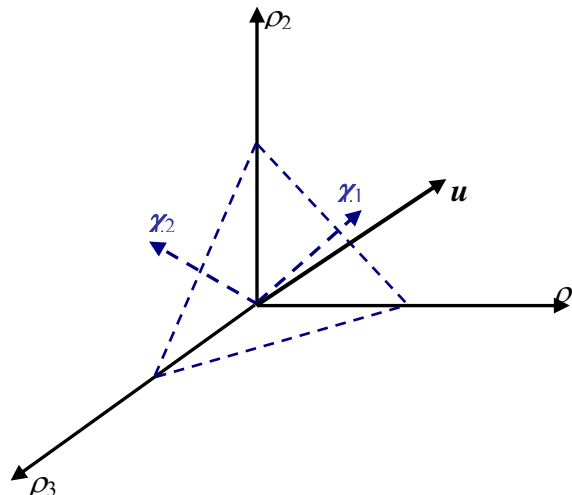


Figura 5.51. Espacio cromático logarítmico 2D. En azul se muestra el plano en el que se define el espacio de color logarítmico 2D  $\boldsymbol{\chi} = (\chi_1, \chi_2)$  definido según la ecuación (5.26) a partir del espacio logarítmico de color  $\boldsymbol{\rho} = (\rho_R, \rho_G, \rho_B)^T$ .

Conocida la dirección de las rectas resultantes al variar la iluminación, es posible obtener una representación 1-D invariante a la iluminación proyectando el vector 2-D  $\boldsymbol{\chi}$  sobre el vector ortogonal a la dirección definida por la iluminación  $\mathbf{e}^\perp$ . Dicha dirección puede obtenerse mediante la calibración en color de las cámaras utilizadas [Finlayson et al. 2002], sin embargo, debido a que en nuestro caso no se ha realizado la citada calibración, el ángulo de proyección de la imagen 2D a la imagen invariante se obtiene como aquel que minimiza la entropía en la imagen 2D [Finlayson et al. 2004].

### 5.3.1. Obtención del ángulo de proyección

En [Finlayson et al. 2004] se propone obtener la imagen invariante mediante la proyección sobre el ángulo para el que el valor de la entropía es mínimo.

En la Figura 5.52 se muestra de forma gráfica esta idea. Se puede observar en la Figura 5.52 (a) que la proyección se ha realizado sobre una línea perpendicular a las rectas que definen los cambios de iluminación sobre el espacio logarítmico de color 2D  $\chi = (\chi_1, \chi_2)$ . Sin embargo en la Figura 5.52 (b) la proyección se ha realizado sobre una recta con una dirección diferente a la perpendicular. Representando el histograma correspondiente en cada caso Figura 5.52 (c) y (d) se puede observar como en el caso de la proyección en la dirección correcta la entropía del histograma es menor.

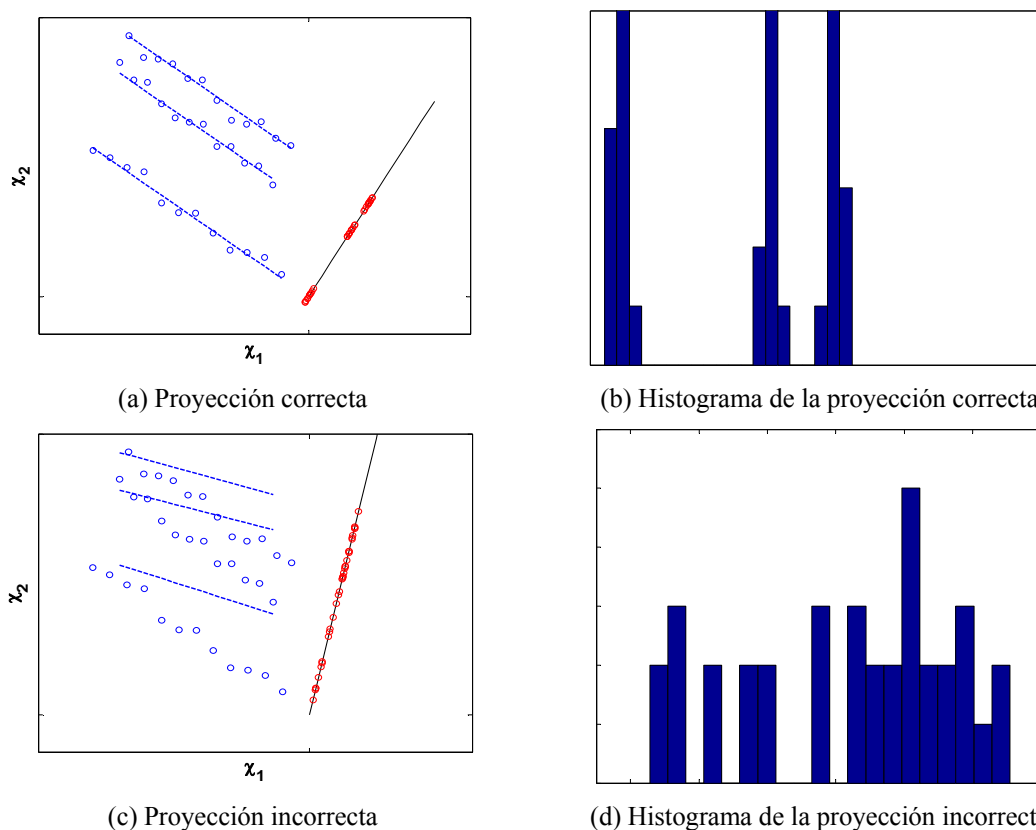


Figura 5.52. Representación de la obtención del ángulo de proyección mediante la minimización de la entropía del histograma de la imagen proyectada. (a) Ejemplo en el que la proyección se ha realizado utilizando el ángulo correcto. (b) Histograma de la proyección correcta. (c) Ejemplo en el que se ha realizado la proyección utilizando un ángulo incorrecto. (d) Histograma de la proyección incorrecta.

Conocido el ángulo  $\theta$ , la proyección del espacio de color  $\chi$  (5.26) a la imagen invariante en escala de grises  $I_{inv}$  se realiza utilizando la expresión (5.27).

$$I_{inv} = \chi_1 \cos \theta + \chi_2 \sin \theta \quad (5.27)$$

Para obtener el ángulo  $\theta$  se proyecta la imagen para todos los ángulos entre 0 y 180° utilizando la ecuación (5.27). Tras la proyección, para el cálculo de la entropía es necesario establecer el rango de los datos en la imagen invariante en escala de grises, así como el número de clases del histograma para cada uno de los ángulos considerados.

Con objeto de eliminar posibles *outliers*, se excluirán aquellos valores que superen el 95% del rango, o se encuentren por debajo del 5%. De esta forma, se empleará únicamente el 90% de los datos para formar el histograma.

En cuanto a la anchura de cada una de las clases, ésta se calcula utilizando la regla de Scott según se indica en [Finlayson et al. 2004].

$$h = 3.5 \cdot \text{std}(\text{datos proyectados}) \cdot N^{1/3} \quad (5.28)$$

donde  $N$  es el tamaño de los datos de la imagen invariante comprendidos entre el 5% y el 95% de los valores del rango de dicha imagen.

Tras calcular la entropía mediante la ecuación (5.29) para cada uno de los ángulos comprendidos entre 0 y 180° se elige para hacer la proyección aquel ángulo para el que el valor de la entropía sea mínimo.

$$\eta = -\sum_i p_i(I_{inv}) \log(p_i(I_{inv})) \quad (5.29)$$

En la Figura 5.53 se muestra la entropía para los ángulos de proyección entre 0 y 180° de una imagen de prueba. En esta figura también se ha señalado el mínimo de la entropía que, en este caso, se produce para un ángulo de proyección de 46°.

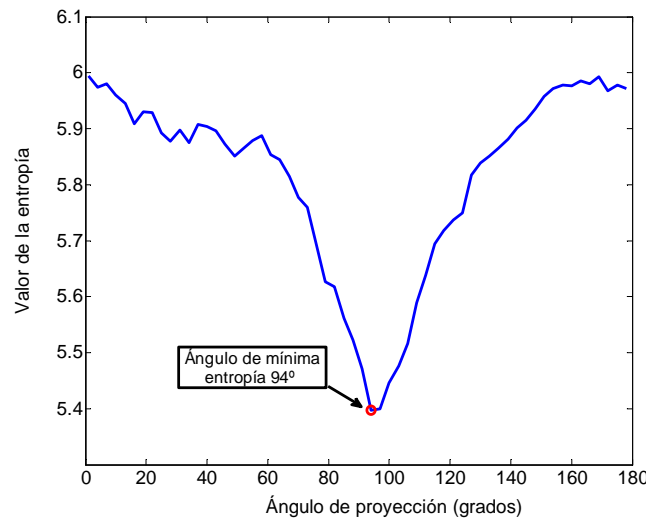


Figura 5.53. Valor de la entropía del histograma de la imagen proyectada (ecuación (5.27)) en función del ángulo utilizado para la proyección. El ángulo de proyección elegido es aquel para el que la entropía es mínima.



### 5.3.2. Detección y eliminación de sombras

Como ya se ha comentado, para la reducción de las sombras se emplea el algoritmo propuesto en [Finlayson et al. 2002] sobre una región de interés (*ROI*) definida alrededor de la curva inicial obtenida tras la umbralización del error de recuperación.

La imagen invariante a la iluminación se obtiene proyectando la *ROI* de la imagen de entrada en la dirección que minimiza la entropía del histograma, obteniéndose el ángulo de proyección, según se ha indicado anteriormente. En la Figura 5.54 se muestra, para una imagen de la secuencia, la región de interés elegida alrededor del robot, así como la imagen invariante a la iluminación, obtenida tras proyectar la imagen de entrada con un ángulo de  $100^\circ$ .



(a) Imagen de entrada. La *ROI* aparece recuadrada en línea roja.

(b) *ROI* de la imagen original.

(c) *ROI* en la imagen invariante a la iluminación.

Figura 5.54. (a) Imagen de entrada en la que se ha representado la región de interés (*ROI*) mediante un rectángulo en rojo. (b) Detalle de la región de interés en la imagen (*ROI*). (c) Imagen invariante a la iluminación obtenida a partir de la *ROI* de la imagen original.

La detección y eliminación de sombras mediante la técnica propuesta en [Finlayson et al. 2002] es posible debido a que, si el ángulo calculado para la proyección es correcto, las sombras desaparecen al transformar la imagen de entrada al espacio invariante a la iluminación. De esta forma, tras extraer los bordes en ambas imágenes (la original en escala de grises y la invariante a la iluminación) se consideran bordes de sombra aquellos que se tienen en la imagen de entrada, y no en la imagen invariante.

Los bordes de cada una de las imágenes se extraen mediante la norma del gradiente. Una vez calculada esta norma se define una función de umbral  $S$  que permite la detección de los píxeles de la imagen en los que se tienen bordes debidos a sombras. Esta función de umbral, dada por (5.30), será igual a uno en aquellos píxeles  $(x,y)$  en los que se cumpla que: la norma al cuadrado del gradiente de la imagen de entrada en escala



de grises ( $\|\nabla \mathbf{I}_g(x, y)\|^2$ ) es superior a un umbral (que denominaremos  $U_g$ ), mientras que la norma al cuadrado del gradiente de la imagen invariante a la iluminación al cuadrado ( $\|\nabla \mathbf{I}_{inv}(x, y)\|^2$ ) es inferior a un segundo umbral ( $U_{inv}$ ). Como ya se ha comentado, aplicando esta función de umbral se obtienen los píxeles correspondientes a zonas de borde que aparecen en la imagen de entrada ( $I_g$ ) y no aparecen en la imagen invariante a la iluminación ( $I_{inv}$ ).

$$S = \begin{cases} 1 & \|\nabla \mathbf{I}_g(x, y)\|^2 > U_g, \|\nabla \mathbf{I}_{inv}(x, y)\|^2 < U_{inv} \\ 0 & \text{resto} \end{cases} \quad (5.30)$$

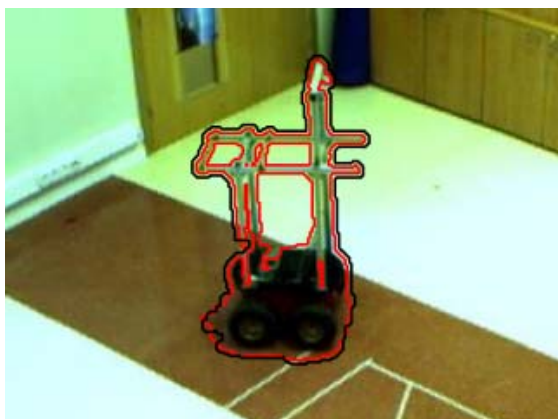
Los valores elegidos para los umbrales  $U_g$  y  $U_{inv}$  en la ecuación (5.30) determinan el resultado del proceso de eliminación de sombras, de forma que es necesario ajustar los valores de estos umbrales de forma correcta.

En la función de umbral mostrada en la ecuación (5.30), la primera condición  $\|\nabla \mathbf{I}_g(x, y)\|^2 > U_g$  debe ser satisfecha por todos los píxeles pertenecientes a zonas de borde en la imagen de entrada en escala de grises  $I_g$ . Es decir, el valor del umbral  $U_g$  debe ser suficientemente pequeño para la correcta detección de todos los bordes de la imagen. Respecto a la segunda condición de la función en (5.30)  $\|\nabla \mathbf{I}_{inv}(x, y)\|^2 < U_{inv}$ , el umbral  $U_{inv}$  debe permitir discriminar entre los bordes correspondientes a sombras, y los debidos a objetos rígidos. Es decir, el valor de  $U_{inv}$  interesa que sea alto para eliminar los bordes de sombras, pero, por otro lado, interesa que sea bajo para no descartar bordes debidos a objetos rígidos. Por este motivo, es importante alcanzar un valor de compromiso al establecer el valor del umbral  $U_{inv}$ .

En la Figura 5.55 se presentan las curvas que definen los contornos de los posibles robots móviles presentes en la escena antes y después del proceso de eliminación de sombras expuesto en este apartado. En la Figura 5.55 (b) se muestra el resultado obtenido al aplicar un umbral  $U_{inv}$  demasiado pequeño. En este caso, apenas se elimina la zona de sombra alrededor del robot. Por otro lado, en la Figura 5.55 (c) se presenta el caso contrario, en el que  $U_{inv}$  es demasiado grande, de forma que se eliminan zonas importantes del robot. Finalmente, en la Figura 5.55 (a) se muestra la curva obtenida al aplicar un umbral  $U_{inv}$  adecuado.



(a)



(b)



(c)

Figura 5.55. Eliminación de sombras. Curvas de contornos iniciales de robots móviles antes (en rojo) y después (en negro) del proceso de eliminación de sombras. (a) Curvas iniciales obtenidas en el caso de aplicar un umbral  $U_{inv}$  adecuado. (b) Curvas iniciales obtenidas en el caso de aplicar un umbral  $U_{inv}$  demasiado pequeño. (c) Curvas iniciales obtenidas en el caso de aplicar un umbral  $U_{inv}$  demasiado grande.

En las imágenes presentadas en la Figura 5.55 se pone de manifiesto la importancia de la correcta elección del valor de los umbrales para la eliminación de sombras. Por este motivo se ha decidido realizar un estudio que permita el correcto ajuste de los valores de los umbrales  $U_g$  y  $U_{inv}$  para la eliminación de sombras en las curvas iniciales.

### 5.3.2.1. Ajuste de los umbrales para la detección y eliminación de sombras

El ajuste de los valores de los umbrales  $U_g$  y  $U_{inv}$  se ha realizado de forma experimental mediante un procedimiento similar al expuesto para el caso del ajuste del umbral sobre el error de recuperación.

En primer lugar se realizó el ajuste supervisado de los valores de  $U_g$  y  $U_{inv}$  para un conjunto de imágenes de prueba pertenecientes a dos secuencias diferentes. A modo de ejemplo, en la Figura 5.56 y la Figura 5.57, se muestran algunas imágenes pertenecientes a cada una de las secuencias.



Figura 5.56. Imágenes de prueba pertenecientes a la secuencia 1. Se muestran las imágenes que ocupan las posiciones 200, 300 y 400 de esta secuencia.

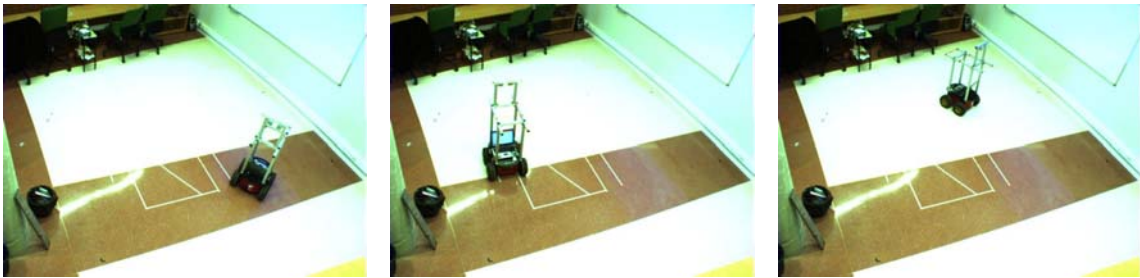


Figura 5.57. Imágenes de prueba pertenecientes a la secuencia 2. Se muestran las imágenes que ocupan las posiciones 200, 300 y 400 de esta secuencia.

El ajuste supervisado de umbrales para el conjunto de imágenes de prueba proporciona, por cada una de estas imágenes, los valores de  $U_g$  y  $U_{inv}$  óptimos, y un conjunto de curvas que definen los posibles contornos de robots móviles presentes en la imagen. Estas curvas no tienen por qué coincidir exactamente con el contorno de los robots, pero son las mejores que se pueden obtener sobre las curvas iniciales obtenidas mediante GPCA utilizando el algoritmo de eliminación de sombras propuesto.

A continuación, en la Tabla 5.14 se muestran los valores de  $U_g$  y  $U_{inv}$  obtenidos para cada una de las imágenes de prueba pertenecientes a las secuencias 1 (Figura 5.56) y 2 (Figura 5.57).

Valor de los umbrales obtenidos de forma supervisada				
Imagen	Secuencia 1		Secuencia 2	
200	$U_g = 30$	$U_{inv} = 0.0030$	$U_g = 30$	$U_{inv} = 0.0037$
250	$U_g = 40$	$U_{inv} = 0.0035$	$U_g = 30$	$U_{inv} = 0.0021$
300	$U_g = 30$	$U_{inv} = 0.0040$	$U_g = 30$	$U_{inv} = 0.0045$
350	$U_g = 30$	$U_{inv} = 0.0040$	$U_g = 30$	$U_{inv} = 0.0020$
400	$U_g = 30$	$U_{inv} = 0.0080$	$U_g = 30$	$U_{inv} = 0.0030$
450	$U_g = 30$	$U_{inv} = 0.0035$	$U_g = 30$	$U_{inv} = 0.0040$
500	$U_g = 30$	$U_{inv} = 0.0037$	$U_g = 30$	$U_{inv} = 0.0025$

Tabla 5.14. Valores de los umbrales  $U_g$  y  $U_{inv}$  obtenidos de forma supervisada para cada una de las imágenes de prueba de las secuencias 1 (Figura 5.56) y 2 (Figura 5.57).

Observando la Tabla 5.14 se aprecia que, mientras un umbral  $U_g = 30$  es adecuado para la mayoría de las imágenes de prueba, el valor del umbral  $U_{inv}$  presenta mayores variaciones. Esto nos permite definir un rango de posibles valores tanto para  $U_g$  como para  $U_{inv}$  para los que se evaluará el funcionamiento del algoritmo de eliminación de sombras. Concretamente, tras observar los resultados reflejados en la Tabla 5.14, los valores elegidos son:

$$U_g = [20, 25, 30, 35, 40, 45, 50]$$

$$U_{inv} = [0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01]$$

Con objeto de determinar los valores de  $U_g$  y  $U_{inv}$ , se han obtenido las curvas iniciales utilizando el proceso de eliminación de sombras para los diferentes valores de ambos umbrales. Para evaluar cuantitativamente la bondad de las curvas obtenidas en cada caso, se han comparado éstas con las estimadas tras el ajuste supervisado de las constantes, para cada una de las imágenes de prueba. En la Figura 5.58 se presentan los resultados obtenidos tanto para las imágenes de la secuencia 1 (Figura 5.56) como para las de la secuencia 2 (Figura 5.57). El valor representado en estas gráficas es el valor medio del porcentaje de píxeles inicializados correctamente para el conjunto de imágenes de la secuencia.

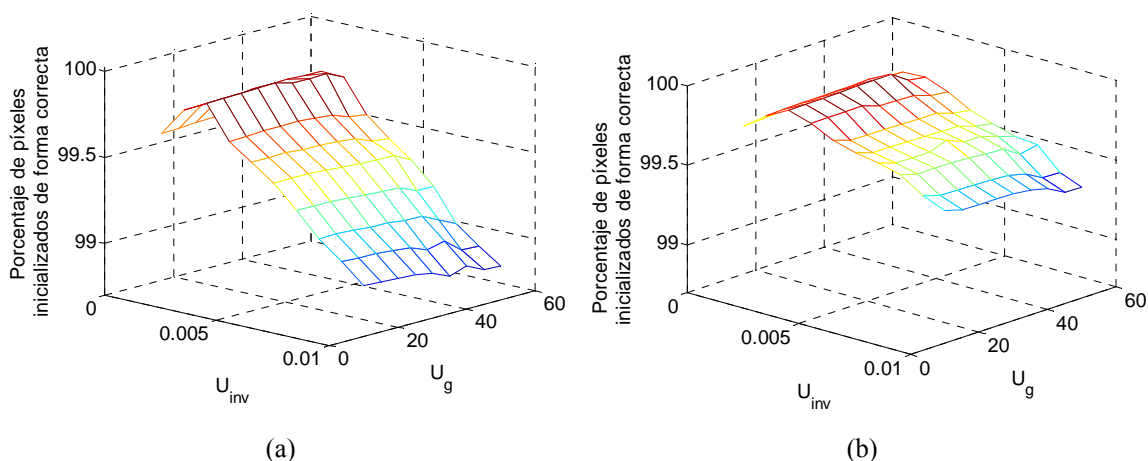


Figura 5.58. Valor medio del porcentaje de píxeles segmentados de forma correcta (a) para el conjunto de imágenes de la secuencia 1 (Figura 5.56) y (b) para el conjunto de imágenes de la secuencia 2 (Figura 5.57).

En las gráficas mostradas en la Figura 5.58 se observa que el valor del umbral  $U_{inv}$  presenta una influencia mucho mayor que  $U_g$ . Esto es así porque mientras la superficie apenas se modifica a medida que se varía  $U_g$ , sí presenta grandes variaciones frente a cambios en  $U_{inv}$ . Además se observa que en ambas gráficas existe un pico alrededor del valor 0.003 del umbral  $U_{inv}$ . Esto también se aprecia en la Tabla 5.15 y Tabla 5.16, en

las que se muestran algunos de los valores representado en las gráficas de la Figura 5.58.

	$U_g = 20$	$U_g = 25$	$U_g = 30$	$U_g = 35$	$U_g = 40$
$U_{inv} = 0.001$	99.72 %	99.73 %	99.73 %	99.73 %	99.73 %
$U_{inv} = 0.002$	99.82 %	99.83 %	99.83 %	99.83 %	99.84 %
$U_{inv} = 0.003$	99.90 %	99.90 %	99.90 %	99.90 %	99.91 %
$U_{inv} = 0.004$	99.85 %	99.85 %	99.82 %	99.81 %	99.81 %
$U_{inv} = 0.005$	99.75 %	99.74 %	99.74 %	99.74 %	99.74 %
$U_{inv} = 0.006$	99.71 %	99.71 %	99.70 %	99.70 %	99.70 %
$U_{inv} = 0.007$	99.68 %	99.68 %	99.72 %	99.67 %	99.67 %
$U_{inv} = 0.008$	99.65 %	99.65 %	99.65 %	99.63 %	99.63 %
$U_{inv} = 0.009$	99.56 %	99.55 %	99.55 %	99.54 %	99.51 %
$U_{inv} = 0.01$	99.50 %	99.50 %	99.50 %	99.49 %	99.46 %

Tabla 5.15. Porcentaje de píxeles segmentados de forma correcta en función de los diferentes valores de los umbrales para la detección de sombras en la imagen  $U_g$  y  $U_{inv}$  considerados para un conjunto de imágenes de la secuencia 1 (a la que pertenecen la imágenes de la Figura 5.56).

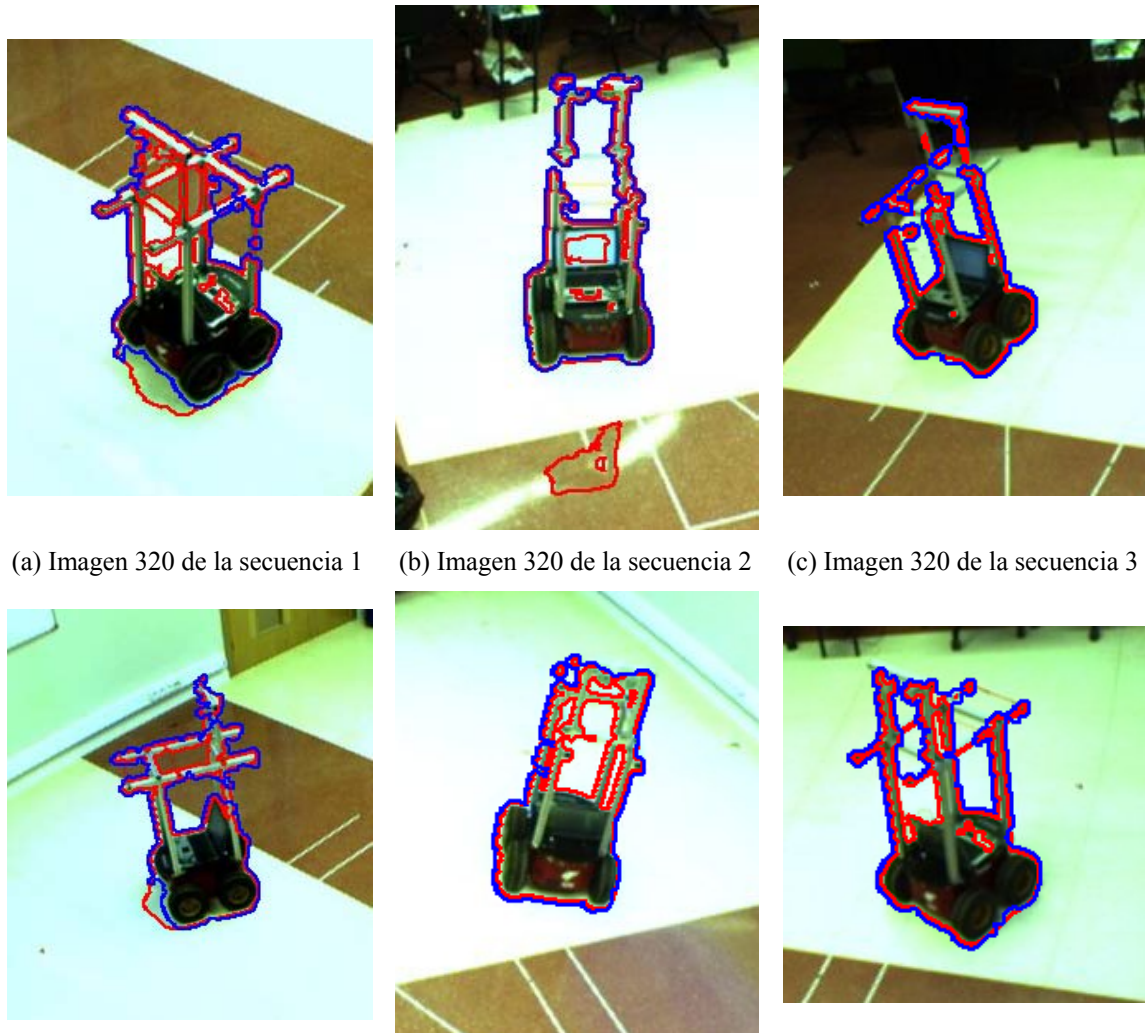
	$U_g = 20$	$U_g = 25$	$U_g = 30$	$U_g = 35$	$U_g = 40$
$U_{inv} = 0.001$	99.62	99.63	99.64	99.65	99.65
$U_{inv} = 0.002$	99.78	99.79	99.80	99.80	99.80
$U_{inv} = 0.003$	99.85	99.86	99.86	99.83	99.83
$U_{inv} = 0.004$	99.66	99.67	99.67	99.66	99.65
$U_{inv} = 0.005$	99.57	99.56	99.56	99.55	99.54
$U_{inv} = 0.006$	99.47	99.47	99.46	99.45	99.45
$U_{inv} = 0.007$	99.34	99.33	99.33	99.32	99.32
$U_{inv} = 0.008$	99.21	99.19	99.19	99.18	99.18
$U_{inv} = 0.009$	99.10	99.09	99.08	99.04	99.06
$U_{inv} = 0.01$	98.98	98.98	98.96	98.92	98.95

Tabla 5.16. Porcentaje de píxeles segmentados de forma correcta en función de los diferentes valores de los umbrales para la detección de sombras en la imagen  $U_g$  y  $U_{inv}$  considerados para un conjunto de imágenes de la secuencia 2 (a la que pertenece la imagen de la Figura 5.57).

A la vista de los resultados mostrados en este apartado se ha decidido establecer los valores de los umbrales:  $U_g = 30$  y  $U_{inv} = 0.003$ . Para evaluar la validez de estos resultados se ejecutó el proceso de eliminación de sombras sobre un conjunto de imágenes pertenecientes a las secuencias 1 y 2 que no habían sido utilizadas para el



ajuste de los umbrales. A modo de ejemplo, en la Figura 5.59 se muestran las curvas iniciales obtenidas para algunas de estas imágenes. Además, también se ha comprobado el correcto funcionamiento de estos umbrales en una serie de imágenes pertenecientes a una tercera secuencia. Las curvas iniciales obtenidas para algunas de estas imágenes se muestran también en la Figura 5.59 (c) y (f).



(a) Imagen 320 de la secuencia 1    (b) Imagen 320 de la secuencia 2    (c) Imagen 320 de la secuencia 3

(d) Imagen 430 de la secuencia 1    (e) Imagen 430 de la secuencia 2    (f) Imagen 430 de la secuencia 3

Figura 5.59. Detalle de las curvas iniciales antes (en rojo) y después (en azul) del proceso de eliminación de sombras propuesto para diferentes imágenes de las secuencias de prueba 1, 2 y 3.

En todos los casos se observa que las curvas iniciales obtenidas aplicando los umbrales indicados en este apartado son adecuadas para la inicialización del algoritmo de segmentación y posicionamiento 3D de robots móviles.

En la Figura 5.60 se muestran los bordes detectados al umbralizar la norma del gradiente de la imagen original en escala de grises (a), los detectados sobre la imagen invariante a la iluminación (b). Se puede observar cómo en la imagen invariante no aparecen los bordes debidos a las sombras en la imagen, pero se mantienen los

correspondientes al robot. Finalmente, en la Figura 5.60(c) se presentan los bordes obtenidos como resultado de aplicar la función de umbral definida en (5.30).

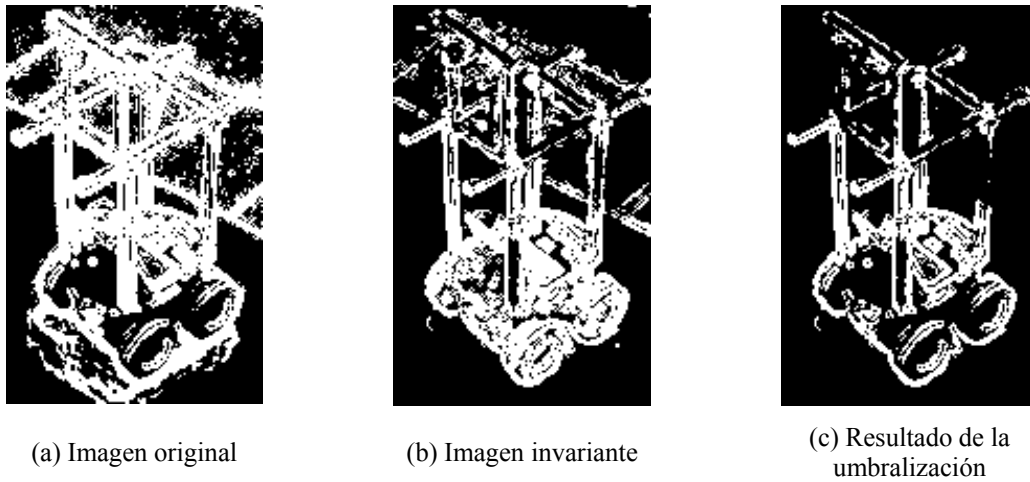


Figura 5.60. (a) Bordes detectados al umbralizar la norma del gradiente de la imagen original. (b) Bordes detectados al umbralizar la norma del gradiente de la imagen invariante. (c) Resultado de la umbralización utilizando la función definida en la ecuación (5.30).

Finalmente, a modo de resumen, en la Figura 5.61 se muestra el diagrama de bloques del proceso completo de eliminación de sombras descrito en este apartado.

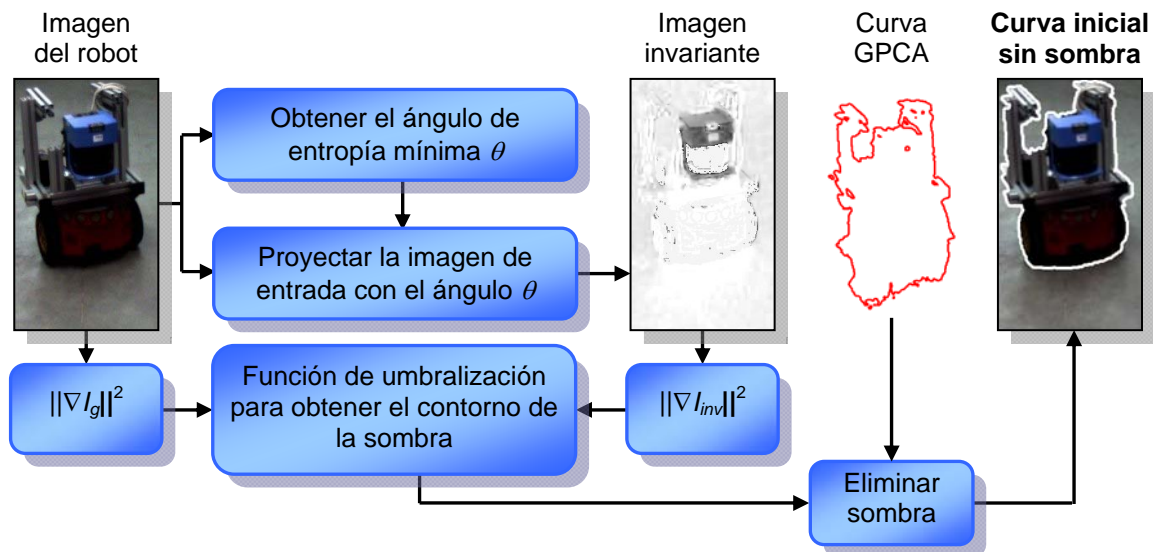


Figura 5.61. Diagrama de bloques del proceso de eliminación de los píxeles que pertenecen a sombras en las curvas iniciales. La eliminación de sombras se realiza en un espacio de color invariante a la iluminación.

Cabe mencionar aquí, que en caso de tener disponibles las imágenes adquiridas por dos o más cámaras calibradas es posible eliminar las sombras sin necesidad de aplicar la técnica descrita en este apartado. Esto se estudia con mayor detalle en el apartado 6.1.4.

### 5.3.3. Inicialización de las curvas sobre las imágenes invariantes a la iluminación

A la vista de los estudios realizados sobre la eliminación de sombras, se planteó la posibilidad de realizar la inicialización y posterior segmentación sobre las imágenes invariantes a la iluminación, en lugar de sobre las imágenes originales en escala de grises. Esto nos permitiría prescindir de la etapa de eliminación de sombras.

En la Figura 5.62 se presenta el esquema del proceso de inicialización sobre imágenes invariantes a la iluminación. Se puede observar que las etapas de este proceso son las mismas que en el caso de imágenes en escala de grises. Sin embargo en este caso se realiza la proyección de las imágenes originales al espacio invariante a la iluminación, y no se requiere realizar la eliminación de sombras tras la umbralización del error de recuperación.

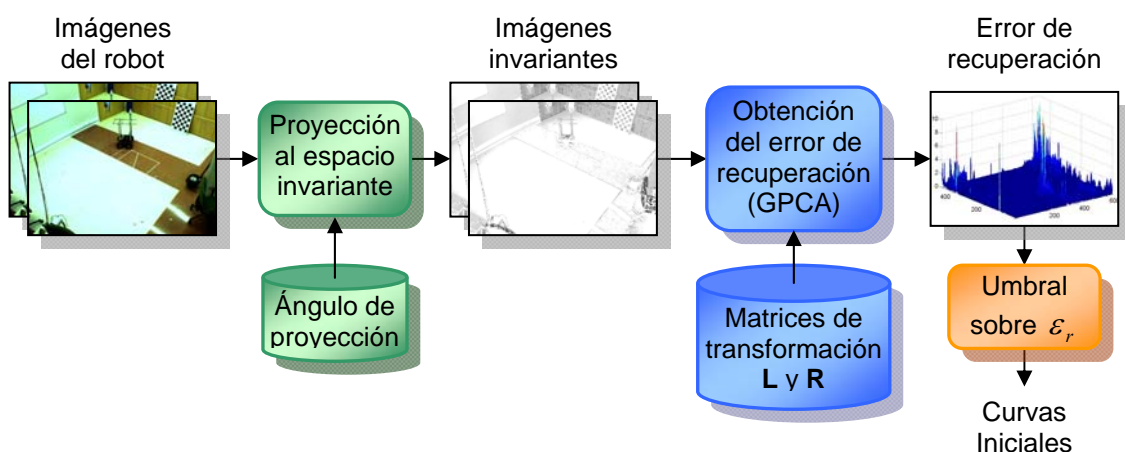


Figura 5.62. Propuesta de inicialización de curvas de contornos de robots, utilizando imágenes invariantes a la iluminación.

Con objeto de comprobar la viabilidad de esta alternativa se han realizado algunos experimentos. En ellos se han comparado los resultados obtenidos al aplicar GPCA sobre las imágenes invariantes a la iluminación, con los que se obtuvieron sobre las imágenes en escala de grises empleando GPCA seguido de un proceso de eliminación de sombras.

Las pruebas realizadas han puesto de manifiesto que, aunque esta representación de las imágenes elimina las sombras en las mismas, provoca que se magnifiquen pequeños cambios en la imagen que pueden introducir errores importantes en la inicialización y posterior segmentación. Por otro lado, se ha comprobado que el tiempo consumido en la



iniciación sobre imágenes invariantes es muy superior al de la iniciación propuesta en esta tesis (GPCA sobre las imágenes en escala de grises, y posterior reducción de las sombras). Debido a que las pruebas experimentales han demostrado la superioridad, tanto en la calidad de las curvas iniciales como en el tiempo de proceso, de la técnica de segmentación propuesta, se ha decidido desechar la iniciación de las curvas en el espacio invariante a la iluminación.



(a) Imagen del fondo invariante a la iluminación    (b) Imagen de entrada invariante a la iluminación    (c) Imagen de entrada en escala de grises

Figura 5.63. Ejemplo de imágenes invariantes a la iluminación. Todas las imágenes pertenecen a la secuencia 1 (Figura 5.19 (a)). (a) Imagen de fondo. (b) Imagen de entrada invariante a la iluminación. (c) Imagen de entrada en escala de grises.

## 5.4. Conclusiones y aportaciones

En este capítulo se ha descrito la solución propuesta en esta tesis para la iniciación de las curvas que definen los contornos de los robots móviles. Además se ha demostrado la importancia de la correcta iniciación de las citadas curvas. Dicha importancia radica en que tanto el resultado de la segmentación como el tiempo consumido en la minimización de la función de energía dependen fuertemente de las curvas iniciales.

El proceso de iniciación de las curvas planteado consta de tres partes bien diferenciadas. Las dos primeras están basadas en GPCA y consisten en la obtención de un modelo del fondo de la escena y la comparación de cada imagen de entrada con dicho modelo. La comparación se lleva a cabo mediante la umbralización del error de recuperación, definido como la diferencia entre la imagen original y la imagen obtenida tras la proyección al espacio transformado GPCA y posterior recuperación de la misma.

Tras las dos primeras etapas se obtiene un conjunto de curvas que se aproximan a los contornos reales de los robots móviles. Sin embargo, debido a las condiciones de iluminación del ISPACE-UAH, los robots proyectan sombras sobre el plano del suelo,

que se incluyen en las curvas iniciales. Estas sombras no se eliminan durante la segmentación de movimiento debido a que se desplazan a la misma velocidad que los robots. Por este motivo, en la inicialización de las curvas se ha incluido una tercera etapa en la que se eliminan las zonas de sombra de las curvas iniciales. La eliminación de sombras se realiza tras proyectar las imágenes a un espacio invariante a la iluminación.

Respecto a las aportaciones, en este capítulo se ha propuesto una solución que permite obtener un conjunto de curvas iniciales que se aproximan a los contornos reales de los robots móviles. Esto permite obtener la segmentación de movimiento y la posición 3D de los robots en un número menor de iteraciones del algoritmo de minimización de la función objetivo, con la consiguiente reducción del tiempo de procesamiento.

Por otro lado, en este capítulo se ha comprobado la importancia de la correcta elección del valor del umbral para el error de recuperación. Se propone, además, un umbral adaptativo en función de la diferencia de la intensidad media de la imagen de entrada y de las imágenes utilizadas para la obtención del modelo de fondo de la escena. El uso del umbral dinámico propuesto incrementa, de forma notable, la robustez del proceso de inicialización de las curvas frente a cambios en la iluminación de las imágenes.

## Capítulo 6

# **Inicialización de la profundidad y estimación del número de robots**



De forma previa a la minimización de la función objetivo definida en la ecuación (4.66) es necesario inicializar tanto las curvas que definen los contornos de los posibles robots móviles en las imágenes captadas por cada una de las  $n_c$  cámaras, como la profundidad de cada punto, también en el sistema de referencia de cada cámara  $\Gamma_{ci}$  ( $i=1, \dots, n_c$ ).

Tras la inicialización de las curvas, es necesario obtener, para cada una de las cámaras disponibles en el espacio inteligente, un valor inicial de la profundidad para cada píxel de las imágenes adquiridas. Además, dado que el algoritmo de segmentación y posicionamiento mediante la minimización de una función objetivo requiere conocimiento a priori del número de robots móviles, es necesario realizar una estimación de este número durante el proceso de inicialización de variables. En este capítulo se detallan las propuestas realizadas en esta tesis referentes a las técnicas utilizadas para la inicialización de la profundidad y la estimación del número de robots móviles.

## 6.1. Inicialización de la profundidad

A la hora de inicializar la profundidad, cabe diferenciar las dos situaciones que se presentan en esta tesis.

En caso de realizar la segmentación de movimiento utilizando la función objetivo definida por Sekkati y Mitiche [Sekkati & Mitiche 2006b], a partir de las imágenes adquiridas por una única cámara, la profundidad se inicializa a un valor constante para todos los puntos. Es decir, se considera que inicialmente se tiene un plano, paralelo al plano imagen, a una distancia  $d_z$  del centro óptico de la cámara. El valor de  $d_z$  puede obtenerse a partir de las características físicas del espacio inteligente.

Por otro lado, en caso de tener múltiples cámaras disponibles, y realizar la segmentación de movimiento y el posicionamiento mediante la minimización de la función objetivo para múltiples cámaras propuesta en esta tesis, la inicialización de la profundidad puede realizarse teniendo en cuenta la información adquirida por todas las cámaras disponibles. En esta línea, se propone el uso del Visual Hull 3D [Laurentini 1991] para la inicialización de la profundidad, a partir de las curvas iniciales obtenidas para las imágenes adquiridas por cada una de las cámaras, mediante GPCA.

En la Figura 6.1 se presentan, en forma de diagrama de bloques, las diferentes etapas implicadas en la inicialización de las curvas y la profundidad para  $n_c$  cámaras. En esta figura se puede observar como las curvas se inicializan de forma independiente para las imágenes captadas por cada una de las cámaras, mientras que la profundidad se estima de forma conjunta para las  $n_c$  cámaras, en el sistema de referencia global  $\Gamma_w$ , y, posteriormente, se proyecta al sistema de referencia de cada cámara.

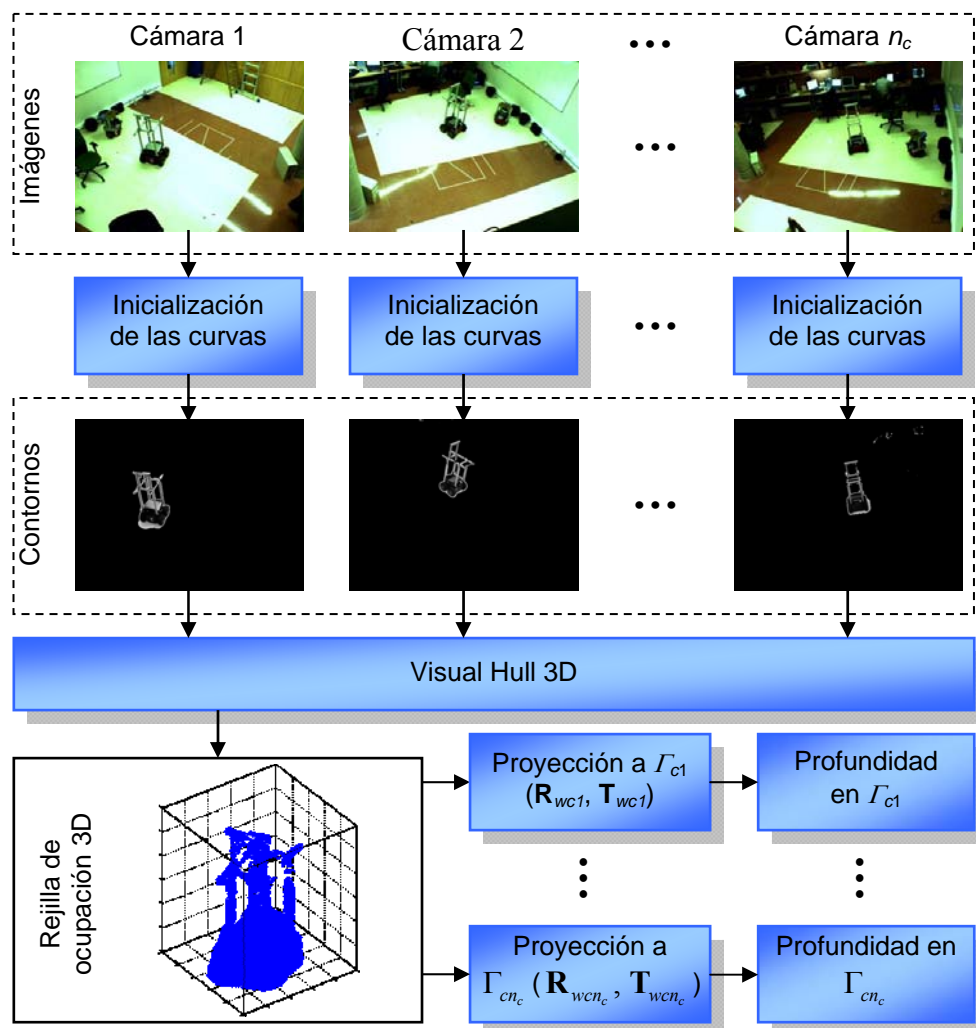


Figura 6.1. Visión general del proceso de inicialización, utilizando Visual Hull 3D de la profundidad a partir de las curvas iniciales que definen los posibles contornos de los posibles robots móviles.

Las curvas iniciales (que definen los posibles contornos de los robots móviles) se obtienen utilizando el algoritmo basado en GPCA y la técnica de eliminación de sombras que han sido descritos a lo largo de los diferentes apartados del capítulo 5. Para ello es necesario obtener las curvas iniciales para las imágenes captadas por cada una de las  $n_c$  cámaras.

La inicialización de la profundidad en cada punto de la imagen, en el sistema de referencia de cada cámara  $\Gamma_{ci}$ , se realiza a partir de los contornos de los robots estimados en la inicialización de las curvas, que son las siluetas que se utilizarán para obtener el Visual Hull 3D. A continuación se presenta el proceso de inicialización de la profundidad utilizando Visual Hull 3D.

### 6.1.1. Visual Hull 3D: definiciones y notación

Partiendo de la suposición de que se tienen  $n_c$  cámaras calibradas, todas ellas captando un objeto 3D, que identificamos como  $O$ , se denomina  $\{S_t^i\}_{i=1}^{n_c}$  al conjunto de siluetas del objeto  $O$  en las imágenes captadas por las  $n_c$  cámaras en el instante temporal  $t$ , y por  $\{C^i\}_{i=1}^{n_c}$  al centro de la cámara  $i$ . Conocidos los parámetros de calibración de las cámaras, es posible obtener la proyección de cualquier punto 3D ( $\mathbf{P}_w$ ) expresado en el sistema de referencia global ( $\Gamma_w$ ) en el plano imagen de cualquiera de las  $n_c$  cámaras. Si se denomina  $\pi^i(\mathbf{P}_w)$  a la función de proyección del punto  $\mathbf{P}_w$  en  $\Gamma_w$  al plano imagen de la cámara  $i$  (es decir,  $p_i = \pi^i(\mathbf{P}_w)$ ), esta notación puede extenderse de forma que, dado un volumen 3D, que identificamos por  $\mathbf{A}_w$  en  $\Gamma_w$ ,  $\pi^i(\mathbf{A}_w)$  representa la proyección del volumen  $\mathbf{A}_w$  sobre el plano imagen de la cámara  $i$  ( $i = 1, 2, \dots, n_c$ ).

Dado un conjunto de  $n_c$  siluetas  $\{S_t^i\}_{i=1}^{n_c}$  y funciones de proyección  $\{\pi^i\}_{i=1}^{n_c}$ , se dice que un volumen  $\mathbf{A}_w$  está definido completamente por las siluetas  $\{S_t^i\}_{i=1}^{n_c}$  si y solo si su proyección sobre el plano imagen de la cámara  $i$ , coincide exactamente con la silueta  $S_t^i$ , ( $i = 1, 2, \dots, n_c$ ). Es decir,  $\pi^i(\mathbf{A}_w) = S_t^i$ . Si existe al menos un volumen no vacío que defina las siluetas exactamente, se dice que el conjunto de siluetas es consistente. En caso contrario, se dice que es inconsistente. Habitualmente, un conjunto de imágenes es consistente, excepto en el caso de que existan errores en la calibración de las cámaras, o ruido en las siluetas.

A continuación se presentan dos definiciones diferentes del Visual Hull en el instante temporal  $t$  ( $H_t$ ) con respecto a un conjunto consistente de siluetas de imágenes captadas por las  $n_c$  cámaras ubicadas alrededor de un objeto 3D,  $O$ , en dicho instante temporal  $\{S_t^i\}_{i=1}^{n_c}$ . A pesar de que ambas definiciones son aparentemente diferentes, ambas son equivalentes, según se demuestra en [German 2003].

### Visual Hull: Intersección de los conos visuales

El Visual Hull  $H_t$  con respecto a un conjunto consistente de siluetas de imágenes  $\{S_t^i\}_{i=1}^{n_c}$  se define como la intersección de los  $n_c$  conos visuales, cada uno de ellos formado por la proyección de la silueta de la imagen  $S_t^i$  al espacio 3D, a través del centro de la cámara  $C^i$ .

Esta primera definición, que es la más utilizada, proporciona un método directo para el cálculo del Visual Hull a partir de las siluetas. Sin embargo, no permite una visión intuitiva del objeto que forman dichas siluetas.

### Visual Hull: Volumen máximo

En general, dado un conjunto de siluetas  $\{S_t^i\}_{i=1}^{n_c}$ , existe un número infinito de volúmenes que encierran las citadas siluetas. La segunda definición define el *Visual Hull*  $H_t$  con respecto a un conjunto consistente de siluetas de imágenes  $\{S_t^i\}_{i=1}^{n_c}$  como el volumen máximo que encierran las siluetas  $\{S_t^i\}_{i=1}^{n_c}$ . Esto implica que el Visual Hull es el volumen convexo máximo que engloba a todas las siluetas, asegurando que el objeto a representar está incluido dentro del Visual Hull.

De esta definición se deduce la primera propiedad fundamental de los Visual Hulls que indica que un objeto  $O$  que forma la silueta  $S_t^i$  se encuentra completamente dentro del *Visual Hull*  $H_t$  construido a partir de  $S_t^i$ .

## 6.1.2. Construcción del Visual Hull

Según la definición primera de *Visual Hull*, dado un conjunto consistente de siluetas, su Visual Hull puede construirse mediante la intersección de los conos visuales. Haciendo esto, el *Visual Hull* se representa en forma de parches sobre una superficie 2D. En la Figura 6.2(a) se muestra un ejemplo en 2D en el que el *Visual Hull* ha sido obtenido mediante la intersección de los conos visuales de 4 cámaras. Esto es sencillo en 2D, sin embargo, esta representación mediante la intersección de los conos visuales es compleja para objetos 3D, ya que el Visual Hull de un objeto 3D general consiste en áreas curvadas e irregulares, difíciles de representar mediante primitivas geométricas simples. El coste computacional y la inestabilidad numérica son otras razones por las que los objetos 3D se aproximan por formas poliédricas [German 2003].



Dado que es complejo intersectar las superficies de los conos visuales en 3D, se utilizan técnicas para simplificar esta tarea. Una de las técnicas más utilizadas es la aproximación basada en vóxeles.

Esta aproximación se basa en el mismo principio de intersección de conos visuales, representando el Visual Hull mediante elementos 3D (vóxeles) en lugar de sobre una superficie 2D. Para ello, el espacio de interés (en nuestro caso el espacio inteligente) se divide en cubos de lado  $\Delta h$ . El conjunto de vóxeles que tienen la misma coordenada  $z$  (por ejemplo  $z=z_i$ ) se denomina “slice”. Aplicando la relación existente entre los vóxeles de un determinado slice, y las imágenes captadas por cada una de las cámaras (dada por las matrices de transformación entre sistemas de referencia  $\mathbf{R}_{wci}$ ,  $\mathbf{t}_{wci}$  y la matriz de parámetros intrínsecos de cada cámara  $\mathbf{K}_{ci}$ ) se obtiene el contorno del objeto ( $O$ ) para cada slice. Repitiendo este proceso para cada uno de los slices en que se ha dividido el espacio, se obtiene finalmente una rejilla de ocupación tridimensional.

A modo de ejemplo, en la Figura 6.2(b) se presenta la aproximación basada en vóxeles en el caso de un objeto 2D. En esta figura se muestra en amarillo el Visual Hull aproximado mediante vóxeles, mientras que la línea verde se corresponde con la intersección real de los conos visuales.

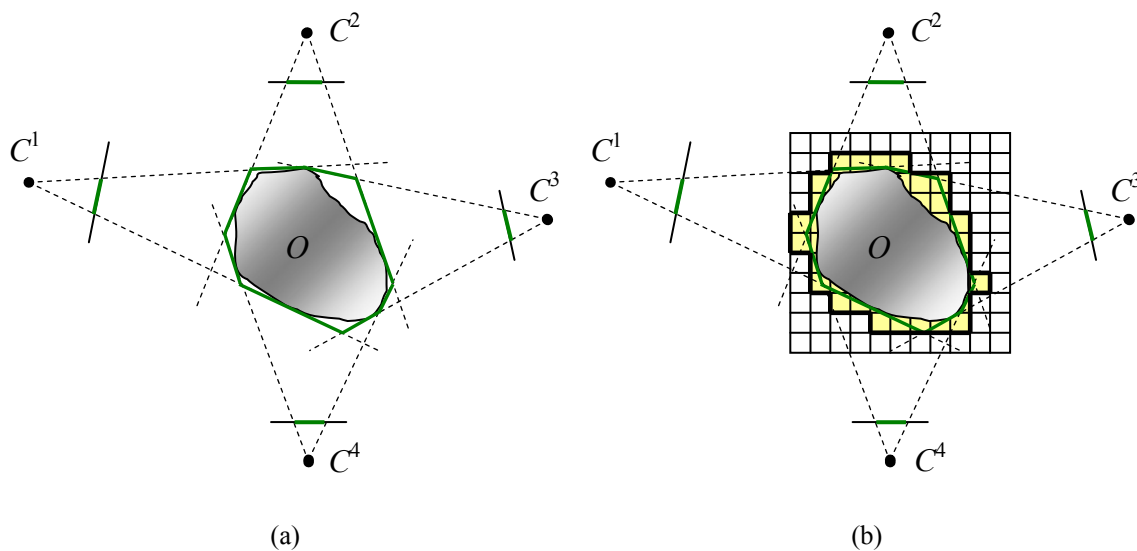


Figura 6.2. Ejemplo de construcción del *Visual Hull* a partir de siluetas en 2D. (a) mediante la intersección directa de los conos visuales (el resultado de la intersección se muestra en color verde alrededor del objeto  $O$ ). (b) Mediante aproximación basada en vóxeles (la región mostrada en amarillo representa el *Visual Hull* aproximado, mientras que la línea verde se corresponde con la intersección real de los conos visuales).

### 6.1.3. Inicialización de la profundidad usando Visual Hull 3D

Como ya se ha explicado en el apartado anterior, usando *Visual Hull 3D* es posible construir una rejilla de ocupación del espacio inteligente formada por cubos de lado  $\Delta h$  a partir de las siluetas definidas por las curvas iniciales para cada una de las  $n_c$  cámaras.

Las posiciones de esta rejilla de ocupación se representan en el sistema de referencia global ( $\Gamma_w$ ). Si se proyectan las posiciones ocupadas al sistema de referencia  $\Gamma_{ci}$  ( $i=1, \dots, n_c$ ) de cada una de las  $n_c$  cámaras mediante las matrices de transformación correspondientes ( $\mathbf{R}_{wci}$  y  $\mathbf{T}_{wci}$ ) se obtienen las coordenadas 3D (en cada  $\Gamma_{ci}$ ,  $i=1, \dots, n_c$ ) de un conjunto de puntos pertenecientes al robot móvil. Esto nos proporciona un método efectivo para la inicialización de la profundidad, a valores próximos a los reales, en el sistema de referencia de cada una de las  $n_c$  cámaras.

Dado que en el sistema de referencia global se incorpora la información de todas las cámaras disponibles, es posible mejorar las curvas iniciales, eliminando zonas que se habían considerado como parte del robot por efecto del ruido o los cambios en la iluminación. Además, se puede reducir el número de puntos para los que se realiza la minimización de energía, con el consiguiente descenso del tiempo de cómputo del algoritmo de segmentación de movimiento y obtención de la posición 3D. A modo de ejemplo, en la Figura 6.3(a) se muestra la rejilla de ocupación obtenida a partir de tres imágenes captadas por las cámaras del ISPACE-UAH. Las curvas iniciales obtenidas para cada una de estas tres imágenes se presentan en la Figura 6.3(b). A partir de la rejilla de ocupación obtenida mediante *Visual Hull 3D* es posible eliminar las curvas que aparecen debidas a brillos, o como en este caso, a movimientos fuera de la zona de interés (ya que, como se ha explicado en la introducción, se considera zona de interés únicamente aquella que captan todas las cámaras consideradas en cada momento). Las nuevas curvas se muestran en la imagen de la Figura 6.3(c).

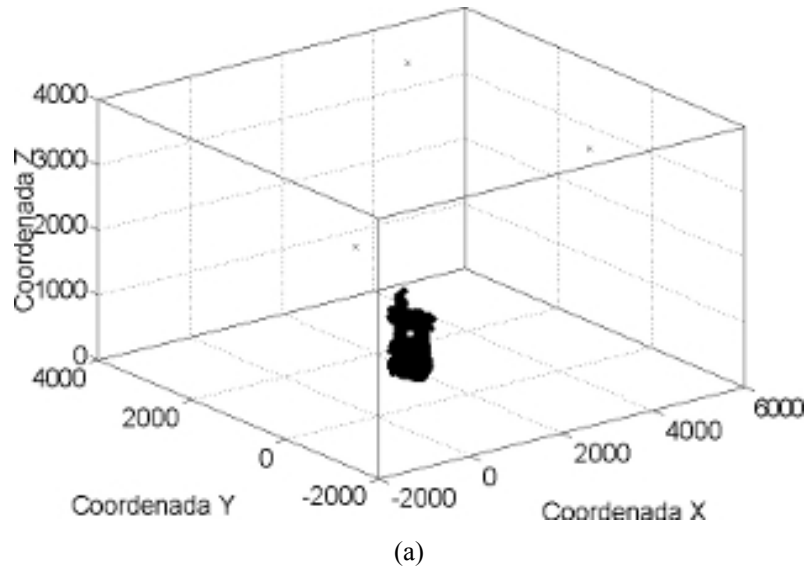


Figura 6.3. Efecto de mejora de las curvas iniciales conseguida tras la inicialización de la profundidad mediante *Visual Hull 3D*. (a) Rejilla de ocupación obtenida a partir de las imágenes captadas por tres de las cámaras disponibles en el ISPACE-UAH. (b) Curva inicial obtenida para la imagen captada por la cámara 3. (c) Curva inicial mejorada tras la inicialización de la profundidad mediante *Visual Hull 3D*. Las curvas que aparecían fuera del robot han sido eliminadas.

Por otro lado, al disponer de información 3D de los robots móviles, es posible reducir de las curvas iniciales las zonas de sombra proyectadas por los robots, eliminando aquellas posiciones de la rejilla de ocupación en las que  $Z_w$  es cero. Es decir, eliminando todos los puntos que se encuentran sobre el plano del suelo según se explica en el siguiente apartado.

#### 6.1.4. Eliminación de sombras en la reconstrucción 3D

En el apartado 5.3 se ha presentado el algoritmo utilizado para la eliminación de sombras mediante la proyección de las imágenes a un espacio invariante a la iluminación. Este método puede aplicarse a cada una de las imágenes captadas por las  $n_c$

cámaras disponibles. Sin embargo, dado que tras el uso de Visual Hull para la inicialización de la profundidad, se tiene información 3D en  $\Gamma_w$ , también es posible utilizar esta información para eliminar la zona de sombra. Esto se hace mediante la eliminación de todos los puntos en los que el valor de la coordenada  $Z_w$  es igual a 0 en el sistema de referencia global  $\Gamma_w$ .

Con objeto de determinar cuál de las dos alternativas proporciona mejores resultados se han realizado diferentes pruebas experimentales y se han comparado los resultados obtenidos al utilizar cada uno de los métodos propuestos. Estas pruebas experimentales se han realizado sobre una secuencia de 100 imágenes en las que aparece un robot móvil. Dichas imágenes han sido adquiridas por cuatro cámaras calibradas, y sincronizadas entre sí. A modo de ejemplo, en la Figura 6.4 se muestra una de las imágenes de la secuencia vista por las 4 cámaras disponibles.

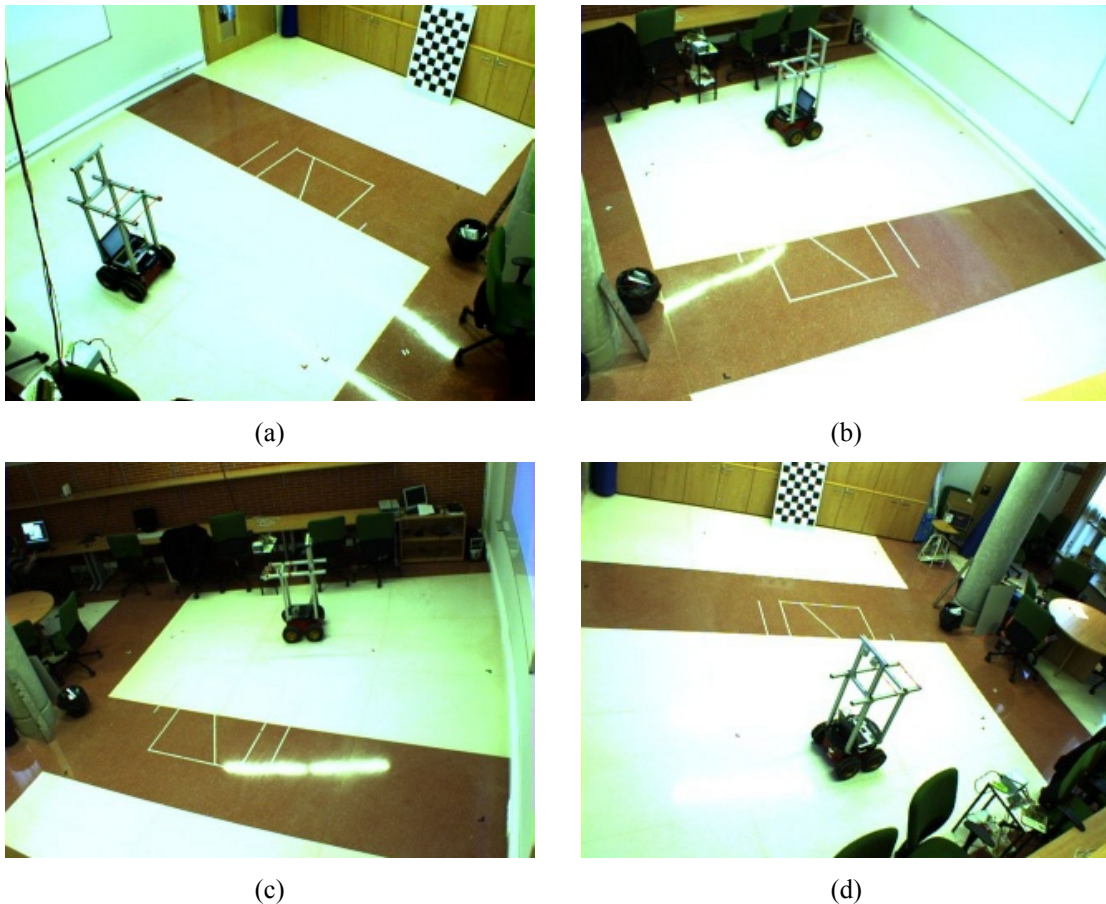


Figura 6.4. Ejemplo de imagen perteneciente a la secuencia de test vista por las cuatro cámaras disponibles en el ISPACE-UAH. (a) Imagen de la cámara 1. (b) Imagen de la cámara 2. (c) Imagen de la cámara 3. (d) Imagen de la cámara 4.

Dado que la calidad de la rejilla de ocupación obtenida mediante *Visual Hull 3D* está relacionada con el número de cámaras disponibles, en primer lugar se analiza el efecto del número de cámaras sobre las curvas obtenidas tras la eliminación de sombras.

En las pruebas experimentales se ha comprobado que, como cabe esperar, a medida que el número de cámaras se incrementa, se obtienen mejores resultados en el proceso de eliminación de sombras. Esto se puede observar en la imagen mostrada en la Figura 6.5(a), en la que se han representado las curvas obtenidas al utilizar las imágenes de 2, 3 y 4 cámaras. En esta figura se puede apreciar como la curva obtenida al utilizar las imágenes de las 4 cámaras se aproxima más al contorno real del robot que la obtenida utilizando únicamente 2. Por otro lado, en caso de utilizar la técnica descrita en el apartado 5.3, las curvas son las mismas, independientemente del número de cámaras, ya que, en este caso, la eliminación de sombras se realiza de forma independiente para cada una de las imágenes de entrada. En la Figura 6.5(b) se puede observar como la curva obtenida es la misma para 2, 3 y 4 cámaras, (la amarilla queda superpuesta a todas las demás curvas en todos los píxeles).

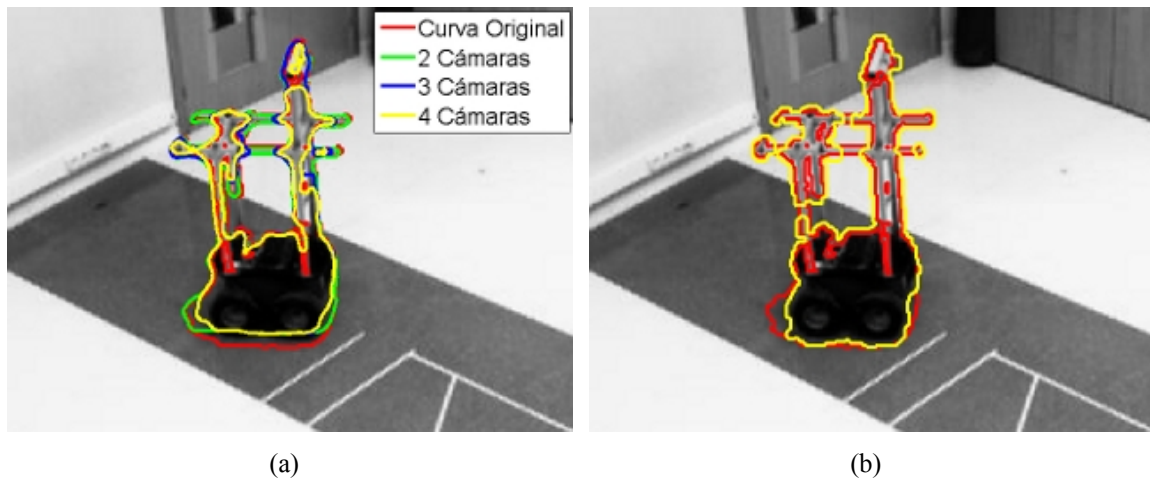


Figura 6.5. Detalle de las curvas de contorno de un robot obtenidas para una imagen de la secuencia de test (Figura 6.4) tras la eliminación de las zonas de sombra. (a) Utilizando *Visual Hull 3D* y eliminando los puntos con  $Z_w=0$  en  $I_w$ . (b) Tras la proyección de la imagen a un espacio invariante a la iluminación.

Con respecto a la “calidad” de las curvas, se ha comprobado que ambas técnicas permiten eliminar la zona de sombra. En la Figura 6.6 se muestran las curvas iniciales obtenidas para diferentes imágenes de la secuencia. En la Figura 6.6(a) se puede observar como ambas técnicas proporcionan resultados similares. Sin embargo, en la imagen mostrada en la Figura 6.6(b), las curvas obtenidas, utilizando el método descrito en el apartado 5.3, se ajustan mejor al robot. Finalmente, en la imagen mostrada en la

Figura 6.6(c) se observa el efecto contrario, siendo mejores los resultados obtenidos tras la eliminación de las muestras en las que  $Z_w$  es nulo.

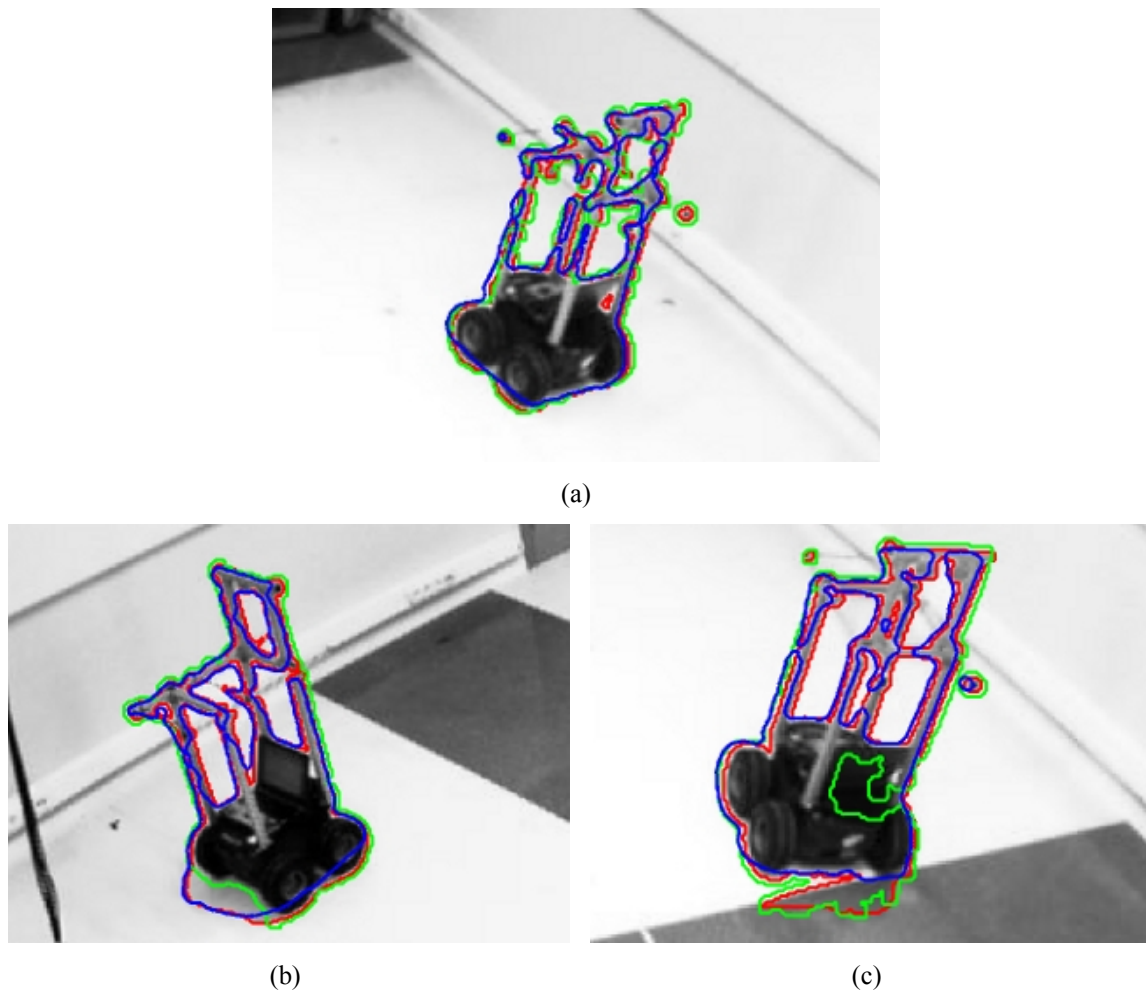


Figura 6.6. Detalle de las curvas obtenidas para diferentes imágenes pertenecientes a la secuencia de test (Figura 6.4). En color rojo se representa la curva inicial obtenida mediante GPCA. La curva verde es la obtenida tras la proyección de la imagen de entrada a un espacio invariante a la iluminación, mientras que la azul es el resultado de eliminar las muestras en las que  $Z_w=0$  en  $I_w$ .

En las pruebas experimentales realizadas, se ha puesto de manifiesto que ambas técnicas permiten eliminar las zonas de sombras de las curvas iniciales. Se ha comprobado, además, que los resultados obtenidos al eliminar las muestras en las que  $Z_w$  es igual a cero, dependen del número de cámaras disponibles. Esta dependencia no se aprecia al utilizar la técnica descrita en el apartado 5.3 ya que la zona de sombra se elimina en las imágenes captadas, de forma independiente, por cada una de las cámaras. Sin embargo, no hay que olvidar que este método hace uso de varios umbrales cuyo ajuste es crítico.

Con objeto de realizar una comparación completa de ambas técnicas, se ha realizado un estudio del tiempo de cómputo consumido al eliminar las sombras



aplicando los dos métodos propuestos: proyectar la imagen a un espacio invariante a la iluminación (según se ha explicado en el apartado 5.3) o eliminar los puntos que pertenecen al plano del suelo en  $\Gamma_w$  ( $Z_w = 0$ ). En la Figura 6.7 se muestra dicho tiempo, en segundos, para 100 imágenes pertenecientes a la secuencia de test (a la que pertenecen las imágenes mostradas en la Figura 6.4) en caso de utilizar las imágenes captadas por 4 cámaras. Los tiempos mostrados en la Figura 6.7 han sido obtenidos al ejecutar los algoritmos en el nodo cliente descrito en el apartado 3.1.1, al presentar la arquitectura hardware del ISPACE-UAH.

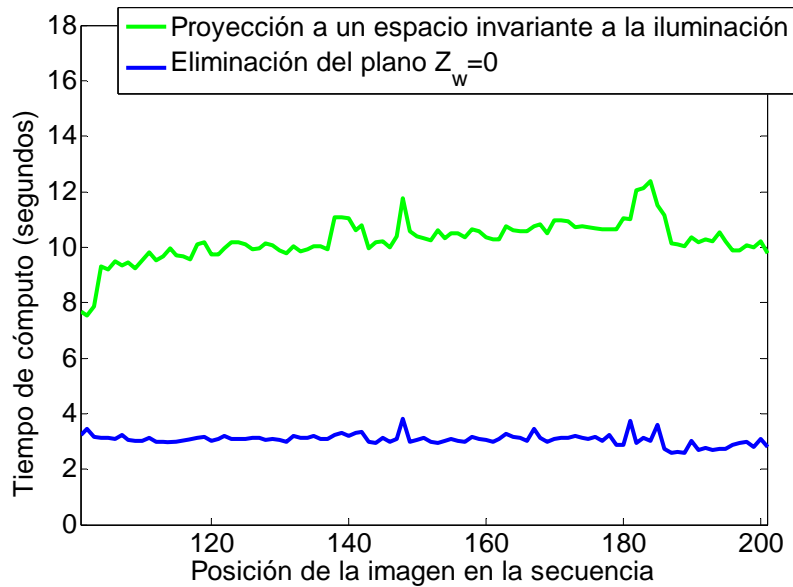


Figura 6.7. Comparación del tiempo de cómputo consumido en la eliminación de los puntos de sombra presentes en las curvas iniciales mediante la proyección a un espacio invariante a la iluminación, y la eliminación del plano  $Z_w=0$  en  $\Gamma_w$  para un conjunto de imágenes pertenecientes a la secuencia de test (Figura 6.4).

En la Figura 6.7 se observa cómo la proyección de las imágenes de entrada a un espacio invariante a la iluminación, para la posterior eliminación de las zonas de sombra en dicho espacio, requiere un mayor tiempo de procesamiento que la eliminación de los puntos del plano  $Z_w=0$  en el espacio de referencia 3D  $\Gamma_w$ . Esto se debe a que la proyección al espacio invariante debe realizarse para cada una de las imágenes captadas por las diferentes cámaras, de forma que el tiempo mostrado es proporcional al número de cámaras. Por otro lado, tras el *Visual Hull 3D*, únicamente se procesa un conjunto de puntos en 3D, que han sido obtenidos a partir de las imágenes de todas las cámaras disponibles.

Debido a la dependencia del tiempo de cómputo con el número de cámaras, se ha repetido el experimento con la misma secuencia de 100 imágenes, considerando 4, 3 y 2 cámaras. El valor medio del tiempo de cómputo obtenido en cada caso se presenta en la Tabla 6.1, donde se puede observar esta fuerte dependencia.

	1 cámara	2 cámaras	3 cámaras	4 cámaras
Proyección a un espacio invariante	0.4817 sg	0.8900 sg	5.4191 sg	10.2422 sg
Eliminación del plano $Z_w = 0$ en $\Gamma_w$	-	2.9585 sg	2.9193 sg	3.0679 sg

Tabla 6.1. Tiempo de cómputo medio (en segundos) consumido en el proceso de eliminación de sombras de las curvas iniciales.

A la vista de los resultados mostrados en la Tabla 6.1, se ha decidido lo siguiente: en caso de utilizar una o dos cámaras, se ejecuta el algoritmo de eliminación de sombras descrito en el apartado 5.3, ya que es el más rápido y que mejores resultados proporciona con 1 o 2 cámaras. Por el contrario, si el número de cámaras es superior a 2, se eliminará el plano  $Z_w = 0$  en el sistema de referencia 3D  $\Gamma_w$  ya que en este caso este algoritmo es más rápido y obtiene mejores resultados.

## 6.2. Estimación del número de robots móviles

El proceso de inicialización de las curvas y eliminación de sombras, que ha sido descrito en los apartados anteriores, proporciona un conjunto de  $N-1$  curvas que dividen el dominio de la imagen en  $N$  regiones. Los puntos interiores a las  $N-1$  curvas son los candidatos a pertenecer a un robot (u otro objeto) móvil. Para finalizar el proceso de inicialización de las curvas, se incluye una última etapa que permite la clasificación del conjunto de puntos candidatos a pertenecer a los robots móviles, así como la estimación del número de objetos móviles presentes en la escena. Esta etapa es necesaria debido a que el algoritmo de segmentación de movimiento propuesto en este trabajo requiere conocimiento a priori del número de robots presentes en la escena. Para llevar a cabo estas tareas se utiliza un algoritmo de agrupación (*clustering*). Este algoritmo se explica con detalle a continuación, en el apartado 6.2.1, analizando por separado los casos en los que se utiliza una o varias cámaras.



## 6.2.1. Algoritmo de clustering

Los algoritmos de *clustering* tratan el problema de la agrupación de datos de acuerdo a determinadas características de los mismos. Una vez realizada la agrupación, cada uno de los conjuntos resultantes se denomina *clase*, o *grupo*, mientras que al conjunto de datos asociados a cada clase se les denomina *miembros de la clase*.

Los algoritmos de agrupación han sido ampliamente utilizados para multitud de aplicaciones, existiendo multitud de alternativas propuestas por diferentes autores. Uno de los métodos de agrupación más ampliamente utilizados es la técnica de las *k*-medias (*k-means*) que es la utilizada en este trabajo.

Con objeto de facilitar la comprensión, a continuación se realiza una breve descripción de la nomenclatura utilizada, del algoritmo *k*-medias y de las modificaciones propuestas para su utilización en esta tesis.

### 6.2.1.1. Nomenclatura

Dado un conjunto  $\mathbf{Y}$  de  $n$  datos ( $y_i$ ) a agrupar  $\mathbf{Y} = \{y_i\}_{i=1}^n$ , representados en el espacio de definición elegido, independientemente de la técnica empleada, el resultado de la agrupación es un conjunto de grupos. En general, el número de grupos se identifica por  $k$  (en nuestro caso  $k$  coincide con número de robots móviles,  $N-1$  en la imagen). Cada uno de estos grupos, que identificaremos por  $\mathbf{G}_j$  ( $j=1, \dots, N-1$ ), se define según la ecuación (6.1).

$$\mathbf{G}_{j=1:N-1} = \left\{ \mathbf{g}_j, \mathbf{I}_{1:l_j, j} \right\}_{j=1}^{N-1} = \left\{ \mathbf{g}_j, \mathbf{L}_j \right\}_{j=1}^{N-1} \quad (6.1)$$

donde  $\mathbf{L}_j = \left\{ \mathbf{l}_{i,j} \right\}_{i=1}^{l_j}$  es un subconjunto de  $\mathbf{Y}$  que representa el conjunto de los  $l_j$  miembros que el algoritmo asocia a la clase  $j$ . Por otro lado  $\left\{ \mathbf{g}_j \right\}_{j=1}^{N-1}$  es el conjunto de vectores característicos, cada uno de los cuales identifica una clase  $j$ , y que habitualmente coincide con el centroide de  $\mathbf{L}_j$  en el espacio de definición elegido.

En cualquier algoritmo de *clustering*, es necesario especificar las características discriminantes que se eligen para la agrupación de los diferentes elementos en  $\mathbf{Y}$ . Esta variable se denomina *vector de características*. El espacio de características más adecuado depende fuertemente de la aplicación y el tipo de datos a agrupar, y de él dependen, en gran medida, la robustez y fiabilidad del algoritmo. En general, el espacio

de características en el que se realiza el agrupamiento tiene un número de dimensiones igual al tamaño del vector de características elegido para el *clustering*.

Una vez definido el espacio de características, el agrupamiento se lleva a cabo empleando una medida de distancia entre el vector característico  $\mathbf{g}_j$  de cada clase  $\mathbf{G}_j$  y cada uno de los datos a agrupar contenidos en  $\mathbf{Y}$ .

### 6.2.1.2. Algoritmo k-medias

El término *k-means* fue usado inicialmente en [Mac Queen 1967] donde se plantean un conjunto de reglas que establecen su funcionamiento:

- Las características elegidas para realizar el *clustering* deben ser lo más discriminantes posible y estar incorreladas.
- Un mismo dato sólo se puede asociar a un *cluster*.
- En caso de que algún cluster quede vacío, se elimina para la siguiente iteración.

En [Lloyd 1982] se propone, por primera vez, el algoritmo estándar de las *k*-medias, aplicado a la cuantificación en la modulación PCM. Otras descripciones del algoritmo básico se encuentran en [Alsabti et al. 1998] o [Kanungo et al. 2002]. En general, el algoritmo *k*-medias, dado un conjunto ( $\mathbf{Y}$ ) de  $n$  datos que se quieren agrupar en  $N-1$  clases, consta de las siguientes etapas:

1. Elegir  $N-1$  datos del conjunto  $\mathbf{Y}$  para utilizarlos como valor inicial de los centroides  $\{\mathbf{g}_{0,j}\}_{j=1}^{N-1}$  de los clusters a crear.
2. Calcular la distancia  $d_{iter,i,j}$  desde cada dato  $\{y_i\}_{i=1}^n$  a cada uno de los centroides  $\{\mathbf{g}_{iter,j}\}_{j=1}^{N-1}$ , y asignar la muestra  $y_i$  a la clase cuyo centroide se encuentre a una distancia menor.
3. Calcular de nuevo los centroides de cada clase  $\{\mathbf{g}_{iter,j}\}_{j=1}^{N-1}$  como la media aritmética del conjunto de datos asignado a dicha clase  $\{\mathbf{L}_{iter,j}\}_{j=1}^{N-1}$ .
4. Comprobar si existe algún cambio entre los elementos asignados a cada cluster respecto a la iteración anterior. Si es así, repetir el algoritmo a partir del paso 2. En caso contrario, eliminar las clases que han quedado vacías y finalizar la ejecución del algoritmo.

Las diferentes etapas descritas se resumen en forma de diagrama de bloques en la Figura 6.8.

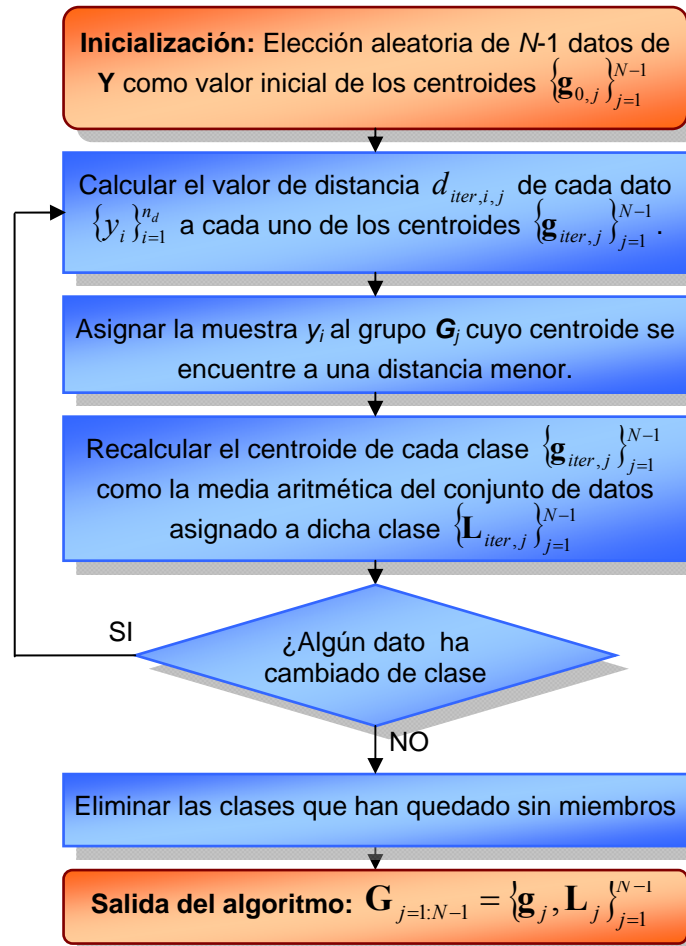


Figura 6.8. Diagrama de bloques general del algoritmo  $k$ -medias básico.

Las principales ventajas del algoritmo  $k$ -medias son su simplicidad a la hora de programarlo, y el bajo tiempo de ejecución por iteración. Sin embargo, el agrupamiento final depende de los centroides elegidos inicialmente. Además, la convergencia a un mínimo global no está garantizada, y el tiempo de cómputo depende del número de iteraciones hasta la convergencia.

Por otro lado, la versión básica del algoritmo  $k$ -medias requiere conocimiento a priori del número de clases. En caso de que dicho número sea desconocido, es necesario modificar el algoritmo  $k$ -medias de forma que sea posible actualizar el número de clases en tiempo de ejecución. La base de este algoritmo es la comparación de las distancias mínimas a los centroides (calculadas para cada dato  $\{y_i\}_{i=1}^n$  utilizando la ecuación (6.2)) con un valor de distancia límite que denominaremos  $dMax$ . En caso de que la distancia

mínima  $d_{\min,iter,i}$  supere la distancia máxima elegida se crea un nuevo grupo y se continua la ejecución del algoritmo de *clustering*.

$$d_{\min,iter,i} = \min(d_{iter,i,j})_{j=1}^{N-1} \quad i = 1, 2, \dots, n \quad (6.2)$$

Las diferentes etapas del algoritmo *k*-medias extendido se muestran en el diagrama de bloques de la Figura 6.9. Los bloques asociados a la creación de una nueva clase aparecen en color amarillo en dicho diagrama.

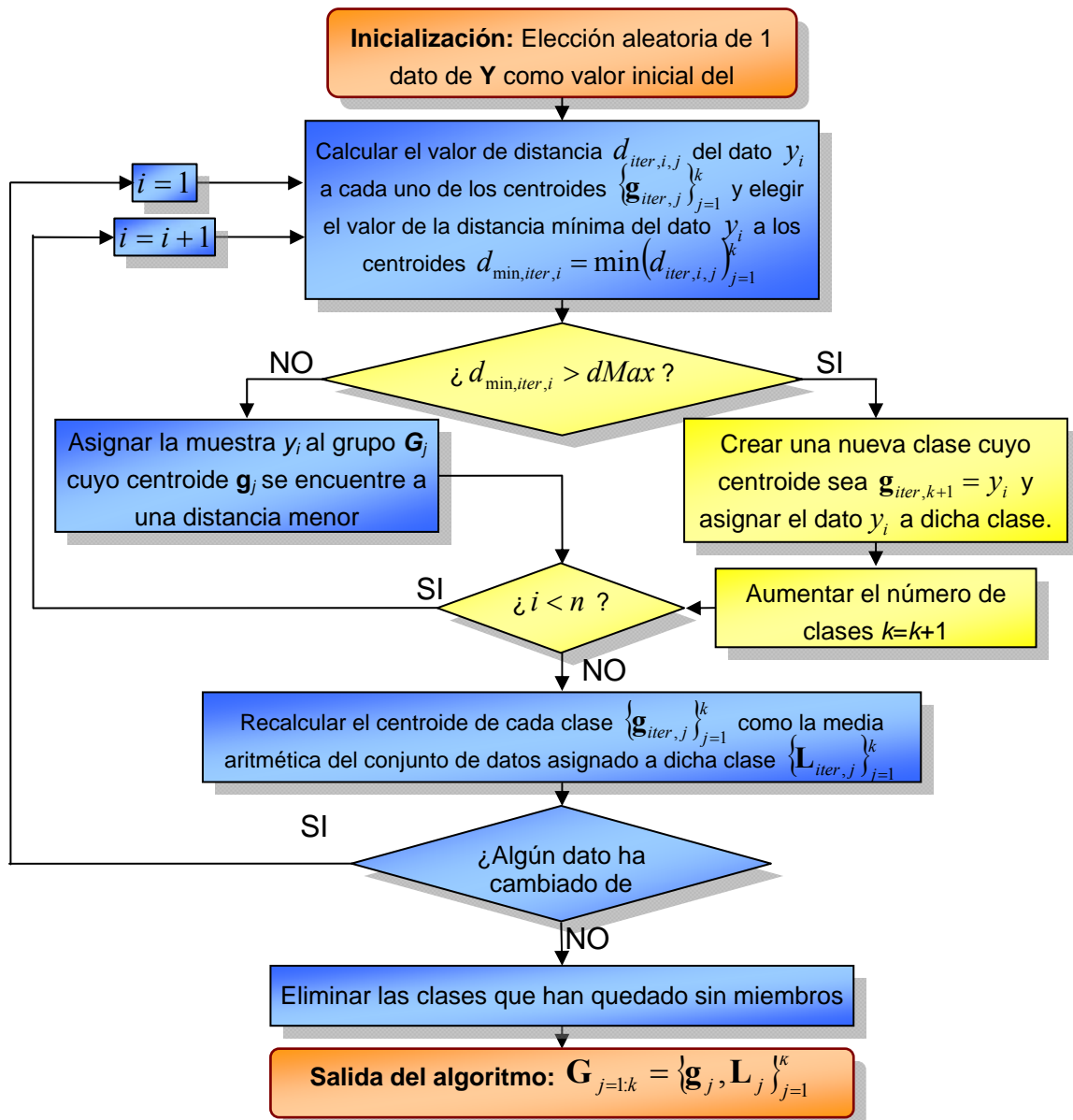


Figura 6.9. Diagrama de bloques general del algoritmo *k*-medias extendido para permitir la creación de clases nuevas. Los bloques añadidos al algoritmo *k*-medias básico aparecen en color amarillo.

A modo de ejemplo, en la Figura 6.10 se muestran los resultados obtenidos al ejecutar el algoritmo de las  $k$ -medias sobre un conjunto de datos  $\mathbf{Y}$  en un espacio 2D, utilizando como características las coordenadas  $x,y$  de los puntos segmentados, y la distancia euclídea como criterio de asignación de elementos a clases. En la Figura 6.10(a) y (b) se presentan los resultados obtenidos al aplicar el algoritmo  $k$ -medias básico con un número de clases de  $k=2$  y  $k=3$ , respectivamente. Mientras que en la Figura 6.10(c) se representa la agrupación obtenida al aplicar  $k$ -medias extendido. En todos los casos, los puntos correspondientes a cada uno de los grupos se han representado en un color diferente, mientras que los centroides se han identificado mediante una cruz negra. Por otro lado, al utilizar  $k$ -medias extendido (Figura 6.10(c)) el número de clases  $k$  no se ha fijado a priori, si bien se ha fijado un valor de distancia límite ( $dMax$ ). Dicho valor está representado mediante las circunferencias centradas en el centroide de cada una de las clases.

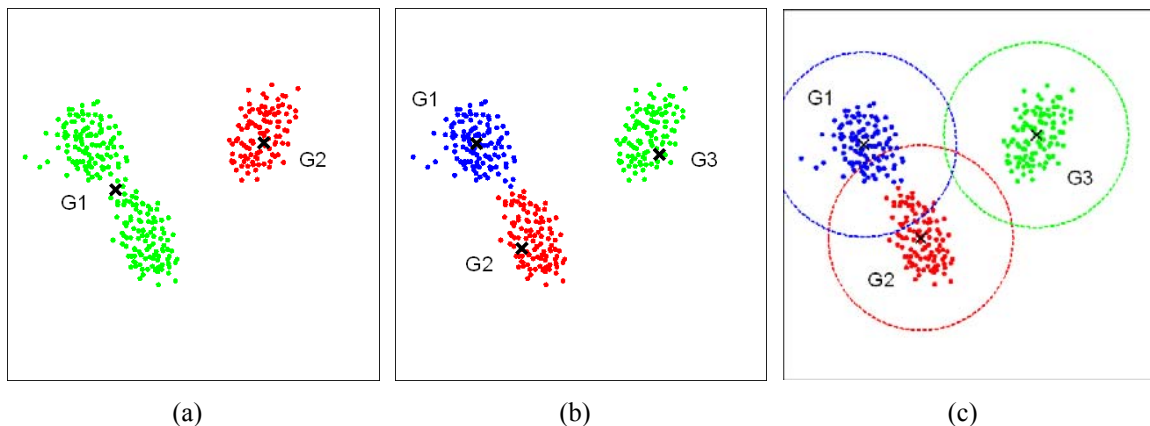


Figura 6.10. Resultados de la ejecución del algoritmo  $k$ -medias sobre un conjunto de datos  $\mathbf{Y}$  en un espacio 2D, utilizando como características las coordenadas  $x,y$  de los puntos segmentados y la distancia euclídea como criterio de asignación de elementos a clases. (a) Resultado obtenido aplicando el algoritmo  $k$ -medias básico con  $k=2$ . (b) Resultado obtenido aplicando el algoritmo  $k$ -medias básico con  $k=3$ . (c) Resultado obtenido aplicando el algoritmo  $k$ -medias extendido. Las circunferencias representan el valor de  $dMax$  en el espacio de clasificación 2D.

Como ya se ha comentado, una de las ventajas del algoritmo  $k$ -medias (tanto en su versión básica como en la extendida) es el bajo tiempo de ejecución por iteración, sin embargo, dado el carácter iterativo de estos algoritmos el tiempo de ejecución final no está acotado, ya que depende del número de iteraciones. Además, al igual que en el caso del  $k$ -medias básico, la convergencia del  $k$ -medias extendido no está garantizada, de forma que es necesario imponer un número máximo de iteraciones permitidas, de forma que al alcanzar este número se da por finalizada la ejecución del algoritmo.

Por otro lado, resulta evidente la alta sensibilidad de esta técnica frente al valor de distancia máxima ( $dMax$ ) elegido, ya que dicho valor determina tanto el número de clases, como el conjunto de elementos de  $\mathbf{Y}$  asociado a cada uno de los grupos  $L_j$  (y consecuentemente el centroide  $g_j$ ). El efecto de  $dMax$  sobre estos factores puede apreciarse en la Figura 6.11, en la que se muestra el resultado del *clustering* obtenido tras ejecutar *k-medias* extendido para tres valores diferentes de  $dMax$ , sobre el mismo conjunto de muestras que en el caso de la Figura 6.10. Es decir, la fiabilidad del sistema propuesto depende fuertemente del valor elegido para este parámetro. En el apartado 6.2.2.3 se realizará un análisis más exhaustivo del efecto del valor de la distancia límite  $dMax$  sobre los resultados obtenidos, además, se realizará un estudio con objeto de determinar cuál es el valor más adecuado para este parámetro en el caso bajo estudio.

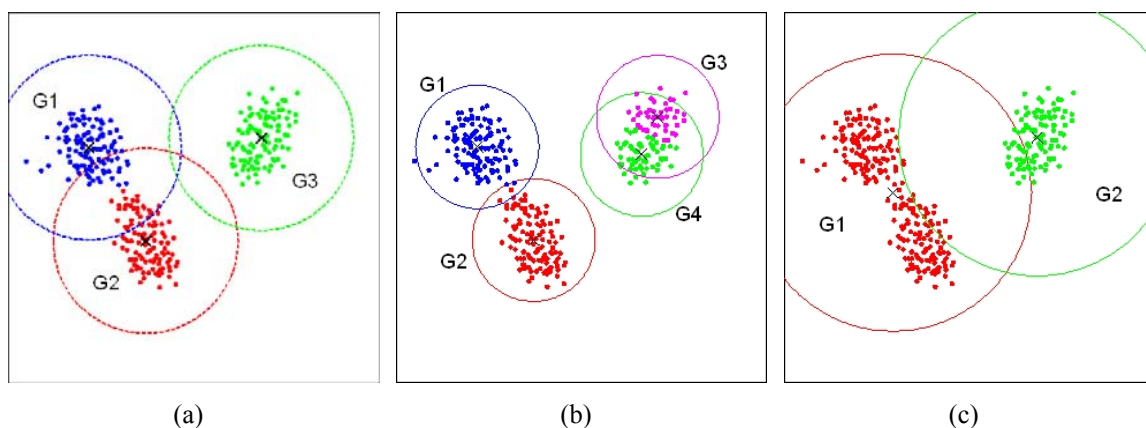


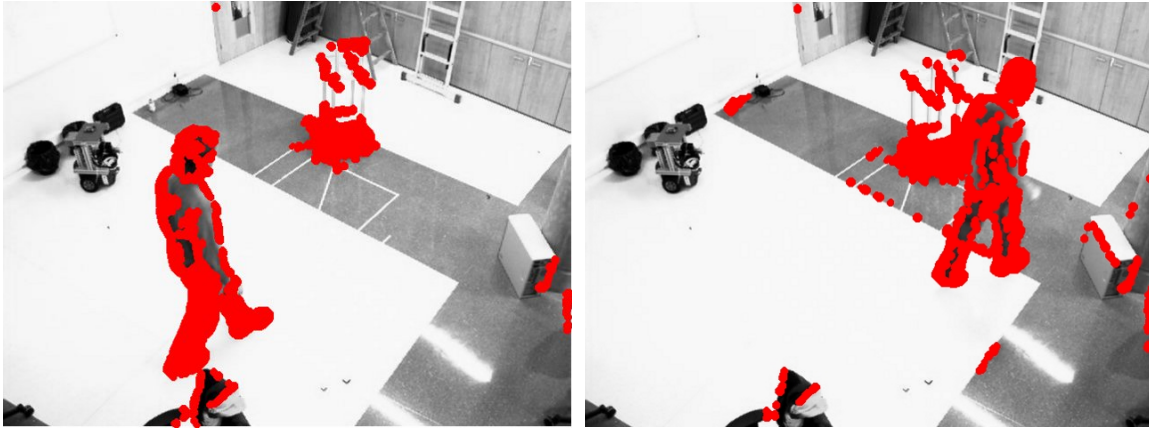
Figura 6.11. Resultados de la ejecución del *k-medias* extendido sobre un conjunto de medidas en un espacio bidimensional para diferentes valores del parámetro  $dMax$ . (a)  $dMax = dMax_0$  las tres clases se detectan correctamente. (b)  $dMax = dMax_0 / 1.5$  una de las clases aparece duplicada. (c)  $dMax = 1.5 \cdot dMax_0$  dos de las clases aparecen unidas.

## 6.2.2. Estimación del número de robots móviles sobre el plano imagen

En caso de que se cuente con una única cámara, la estimación del número de robots móviles se realiza sobre el plano imagen. En los siguientes apartados se describe el algoritmo empleado para la estimación del número de robots móviles y la separación de las curvas iniciales sobre el plano imagen. Previamente, es necesario definir la característica empleada para discriminar entre los diferentes grupos, así como el espacio de características (en el que se representa el vector de características).

### 6.2.2.1. Espacio de características

Tras realizar la inicialización de las curvas mediante GPCA y eliminar las zonas de sombra, el resultado es un conjunto de píxeles sobre el plano imagen, que son candidatos a pertenecer a robots móviles. En la Figura 6.12 se muestran dos imágenes pertenecientes a la secuencia 4 (la misma a la que pertenecen las imágenes mostradas en la Figura 5.48(d), (e) y (f) sobre las que se muestran los puntos candidatos a pertenecer a robots móviles obtenidos tras el proceso de inicialización de las curvas.



(a) Imagen 21 de la secuencia 4

(b) Imagen 50 de la secuencia 4

Figura 6.12. Imágenes pertenecientes a la secuencia 4 (Figura 5.48d) sobre las que se muestran los puntos candidatos a pertenecer a robots móviles, obtenidos tras la inicialización de las curvas mediante GPCA y eliminación de sombras.

Dado que el conjunto de medidas (coordenadas de los píxeles candidatos a pertenecer a robots móviles) son sobre el plano imagen 2D de la cámara, este es el espacio elegido para definir el vector de características. Por tanto los vectores de características serán bidimensionales, siendo sus componentes las coordenadas píxelicas  $(x_i, y_i)$  sobre el plano imagen.

Una vez definido el espacio de características, es necesario elegir el criterio de clasificación para la asignación de elementos a *clusters*, así como ajustar el valor de distancia límite  $dMax$ . Dado que ambos parámetros están directamente relacionados entre sí, el ajuste de los mismos se aborda a continuación.

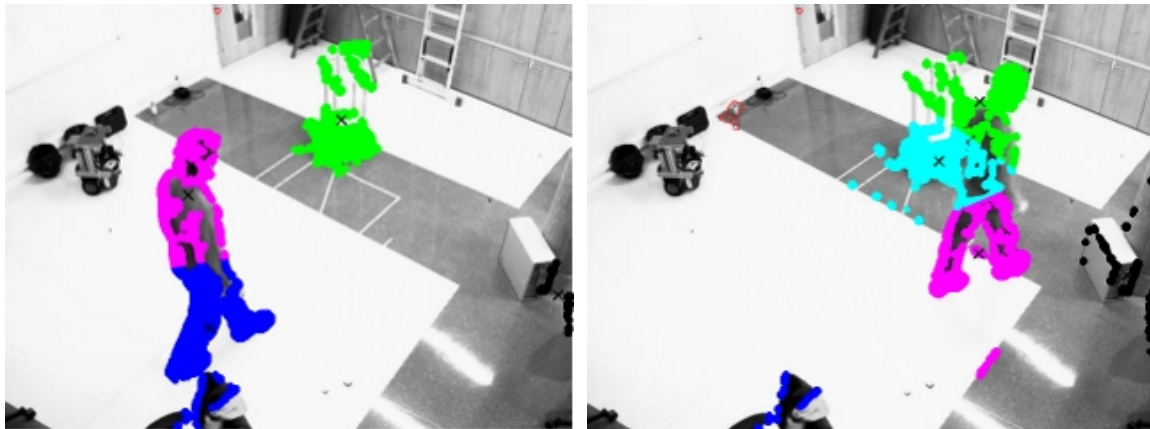
### 6.2.2.2. Medida de la distancia

A la hora de definir la medida de distancia para la asignación de elementos a *clusters* dentro del proceso del algoritmo *k-medias*, la alternativa más sencilla es el uso de la distancia euclídea presentada en (6.3).

$$d(\mathbf{x}, \mathbf{g}) = \sqrt{(x_i - g_x)^2 + (y_i - g_y)^2} \quad (6.3)$$



Sin embargo, en las imágenes mostradas en la Figura 6.12 se puede observar que, debido a las características de los robots y los usuarios del sistema, las dimensiones de los objetos móviles en los ejes  $x$  e  $y$  del plano imagen difieren notablemente entre sí. Por tanto, si la medida elegida es la distancia euclídea (ecuación (6.3)) entre las coordenadas del píxel  $(x_i, y_i)$  sobre el plano imagen, y el centroide  $\mathbf{g} = (g_x, g_y)$  de cada una de las clases, se observa una incorrecta segmentación debido a que en este caso las dimensiones de los objetos en la dirección del eje  $y$  son mucho más grandes que en la dirección del eje  $x$ . En la Figura 6.13 se muestra un ejemplo en el que se ha ejecutado  $k$ -medias sobre los datos de entrada obtenidos para las imágenes mostradas en la Figura 6.12, utilizando como medida la distancia euclídea, y con un valor de distancia límite  $dMax = 150$ . En esta figura se puede observar la incorrecta segmentación debido a la disparidad en las dimensiones de los objetos en la dirección de los ejes  $x$  e  $y$ .



(a) Imagen 21 de la secuencia 4

(b) Imagen 50 de la secuencia 4

Figura 6.13. Resultados obtenidos tras aplicar  $k$ -medias sobre los datos de entrada mostrados en la Figura 6.12 utilizando como medida la distancia euclídea, con un valor de distancia límite  $dMax = 150$ .

Con objeto de reducir, en la medida de lo posible, los defectos de clasificación debidos a la disparidad en las dimensiones de los objetos en la dirección de los ejes  $x$  e  $y$ , se proponen dos alternativas que se explican a continuación. En cada uno de los casos es necesario realizar un estudio que permita obtener el valor de distancia límite ( $dMax$ ) más adecuado para su aplicación en esta tesis.

A la vista del problema de clasificación mencionado (Figura 6.13) se propone utilizar un vector de características que incluya solamente la coordenada  $x$  de cada uno de los píxeles. Por tanto, la medida de distancia entre cada medida y el centroide de cada clase es la mostrada en la ecuación (6.4).

$$d(\mathbf{x}, \mathbf{g}) = \|x_i - g_x\| \quad (6.4)$$



En la Figura 6.14 se muestra el resultado de la proyección sobre el eje  $x$  de las medidas mostradas en la Figura 6.12(a) y (b).

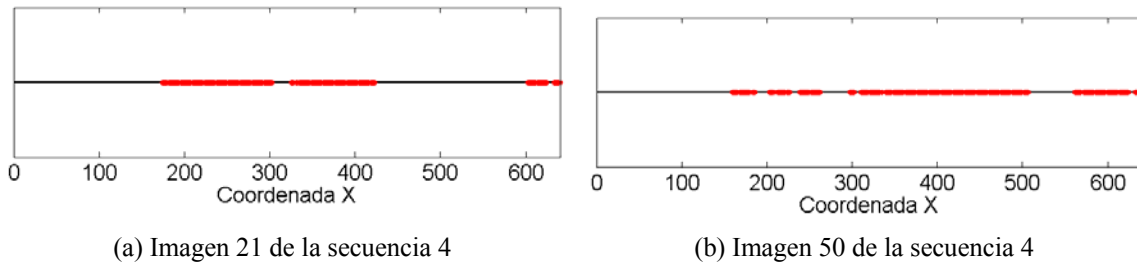


Figura 6.14. Resultado de la proyección sobre el eje  $x$  de las medidas mostradas en las imágenes de la Figura 6.12(a) y (b).

Respecto a la segunda alternativa, esta se basa en la ponderación de la contribución de las distancias en cada uno de los dos ejes del sistema de coordenadas del plano imagen mediante un factor multiplicador de la distancia en el eje  $y$ . De esta forma, la medida de distancia utilizada es la mostrada en la ecuación (6.5), donde  $\tau$  es una constante que pondera la contribución de la distancia en el eje  $y$  respecto a la distancia en el eje  $x$  sobre el plano imagen. El valor de  $\tau$  está determinado por la relación entre las dimensiones en los ejes  $x$  e  $y$  de los robots (u otros elementos móviles) sobre el plano imagen.

$$d(\mathbf{x}, \mathbf{g}) = \sqrt{(x_i - g_x)^2 + (\tau(y_i - g_y))^2} \quad (6.5)$$

El cualquiera de los casos anteriores, el valor elegido para la  $dMax$  determina la calidad de los resultados obtenidos. Por este motivo, se ha realizado un estudio exhaustivo para determinar el valor de distancia límite más adecuado al utilizar cada una de las medidas de distancia propuestas (mostradas en las ecuaciones (6.3), (6.4) y (6.5)). Tanto las pruebas experimentales realizadas, como los resultados obtenidos se presentan en el siguiente apartado.

### 6.2.2.3. Ajuste del valor de la distancia límite

El valor de distancia límite,  $dMax$ , está directamente relacionado con el tamaño de los robots y otros elementos en el ISPACE-UAH al ser proyectados sobre el plano imagen. Por este motivo, para fijar un valor inicial alrededor del cual realizar el ajuste se ha decidido calcular la distancia máxima entre los puntos pertenecientes a un cubo de dimensiones 50x50x100 (largo x ancho x alto) centímetros al ser proyectado del sistema de referencia del mundo  $\Gamma_w$  al plano imagen. Esta distancia se ha calculado utilizando las tres ecuaciones descritas en el apartado anterior. En el caso de la distancia euclídea

ponderada (6.5), ha sido necesario fijar también el valor de la constante de ponderación  $\tau$  que depende de la relación entre las dimensiones en los ejes  $x$  e  $y$  de los robots sobre el plano imagen. Para estimar el valor de  $\tau$  se ha considerado también que los robots son cubos de dimensiones 50x50x100 centímetros en  $\Gamma_w$ . Bajo este supuesto, se ha obtenido una relación entre la distancia máxima en el eje  $y$  y la distancia máxima en el eje  $x$  del orden de 2/3. Los resultados obtenidos para las diferentes alternativas para la medida de la distancia se muestran en la Tabla 6.2.

Es evidente que estos valores son únicamente válidos para la cámara considerada en este estudio. Sin embargo, dada la distribución de las cámaras en el ISPACE-UAH, los resultados obtenidos aquí son fácilmente extrapolables para el resto de cámaras.

Medida de distancia utilizada		Distancia máxima (d0) en milímetros
<b>Distancia Euclídea</b>	$d(\mathbf{x}, \mathbf{g}) = \sqrt{(x_i - g_x)^2 + (y_i - g_y)^2}$	213,53
<b>Distancia Euclídea ponderada (<math>\tau=1/3</math>)</b>	$d(\mathbf{x}, \mathbf{g}) = \sqrt{(x_i - g_x)^2 + (\tau(y_i - g_y))^2}$	142,93
<b>Distancia sobre el eje X</b>	$d(\mathbf{x}, \mathbf{g}) = \sqrt{(x_i - g_x)^2}$	135,91

Tabla 6.2. Valor de la distancia máxima entre los píxeles obtenidos tras proyectar un cubo de dimensiones 50x50x100 centímetros del sistema de referencia del mundo  $\Gamma_w$  al plano imagen.

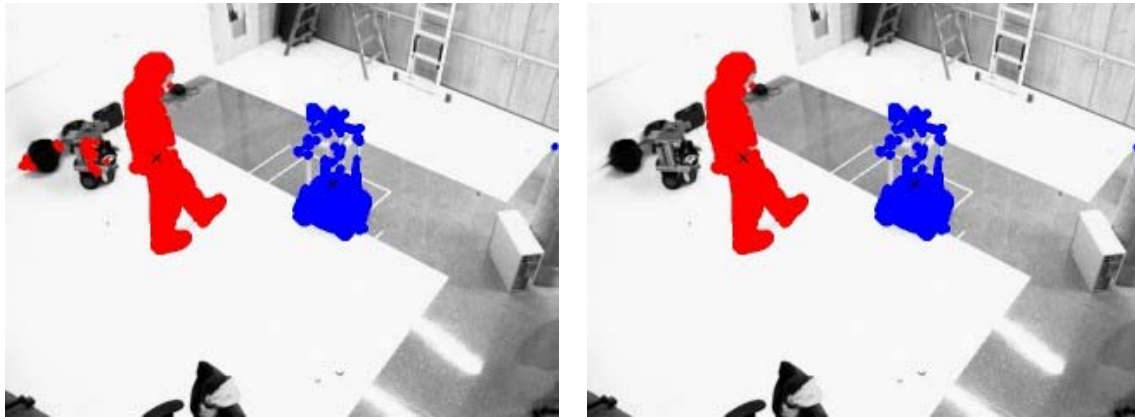
A partir de los valores mostrados en la Tabla 6.2 se ha decidido evaluar los resultados del clustering asignando a  $dMax$  un conjunto de valores alrededor de la distancia máxima mostrada en la tabla. En concreto, por cada una de las alternativas propuestas para la medida de la distancia se han realizado pruebas experimentales sobre 150 imágenes pertenecientes a la secuencia compleja, concretamente a la secuencia 4 (Figura 5.48(d)) en la que aparece tanto un robot móvil, como un usuario. Los diferentes valores de  $dMax$  considerados en esta prueba se han obtenido multiplicando la distancia máxima mostrado en la Tabla 6.2 por un valor comprendido entre 0.70 y 1.30. Esto permite realizar pruebas con valores de  $dMax$  que se encuentran tanto por encima como por debajo del valor mostrado en la Tabla 6.2. En la Tabla 6.3 se muestra el porcentaje de errores de unión y duplicación de clases obtenidos en cada uno de los casos.

Factor de multiplicación		0.70·d0	0.85·d0	1·d0	1.15·d0	1.30·d0
Distancia euclídea	Unión de clases	0 %	4.67 %	4.67 %	9.33 %	16 %
	Duplicación de clases	95.33 %	56.00 %	56.00 %	27.33 %	6.67 %
Distancia euclídea ponderada	Unión de clases	0 %	0 %	0 %	4.67 %	4.67 %
	Duplicación de clases	69.33 %	67.33 %	64.00 %	71.33 %	29.33 %
Distancia en el eje X	Unión de clases	0 %	0 %	0 %	4.67 %	4.67 %
	Duplicación de clases	69.33 %	69.33 %	64.67 %	36.67 %	30.67 %

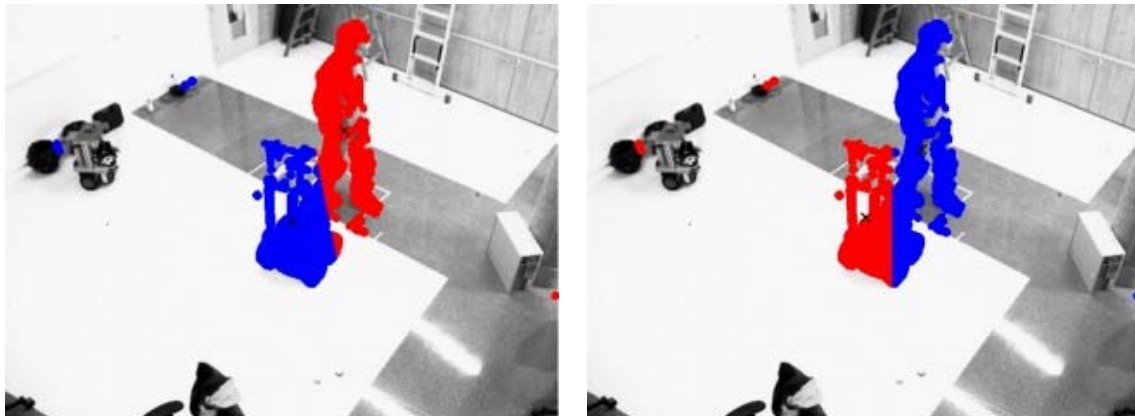
Tabla 6.3. Porcentaje de errores de fusión y duplicación de clases en función del factor de multiplicación utilizado para fijar el valor de la distancia límite. Resultados obtenidos utilizando la distancia euclídea (ecuación (6.3)), la distancia euclídea ponderada con  $\tau=2/3$  (ecuación (6.5)) y la distancia sobre el eje  $x$  (ecuación (6.4)).

A la vista de los porcentajes de error presentados en la Tabla 6.3, como cabe esperar, a medida que se incrementa el valor de  $dMax$  aumenta el porcentaje de errores de unión de clases, mientras que el error de duplicación de clases disminuye. Esto implica que es necesario alcanzar una solución de compromiso en la elección del valor de  $dMax$ .

En este caso se ha elegido un valor de  $dMax = 1.30 \cdot d0$ . Los resultados del *clustering*, para algunas de las 150 imágenes de la secuencia de test, se muestran en la Figura 6.15. En todos los casos, la imagen mostrada en la primera columna refleja el resultado obtenido utilizando (6.5) mientras que la segunda muestra el resultados utilizando (6.4). No se incluye el resultado utilizando (6.3) debido a que, según se puede observar en la Tabla 6.3, el número de errores es muy elevado. En la Figura 6.15 se observa como en los casos en que la persona y el robot se encuentran a una distancia suficientemente grande como en los casos (a) y (c), tanto la utilización de (6.4) como de (6.5) proporcionan buenos resultados. Por otro lado, en la Figura 6.15(b) se observa como la distancia euclídea ponderada permite una mejor discriminación entre el robot y la persona cuando ambos se encuentran próximos entre sí.



(a) Imagen 1 de la secuencia



(b) Imagen 50 de la secuencia



(c) Imagen 100 de la secuencia

Figura 6.15. Resultado del clustering tras aplicar *k-medias* utilizando como medida de distancia la ecuación (6.5) (imágenes mostradas a la izquierda) y la ecuación (6.4) (imágenes mostradas a la derecha) para (a) Imagen 1 de la secuencia.

En cualquiera de los casos, cabe destacar que, mediante el algoritmo *k-medias* únicamente se pretende obtener una estimación del número de objetos móviles presentes en la escena, y un primer agrupamiento de las medidas, ya que la segmentación de movimiento propiamente dicha se realiza mediante la minimización de la función objetivo definida según la ecuación (4.14) en caso de utilizar las imágenes de una única cámara, o la ecuación (4.66) para el caso de múltiples cámaras.

Con objeto de evaluar la validez de las propuestas realizadas para la estimación del número de robots (u otros elementos) móviles presentes en las imágenes de una secuencia, se ha realizado un conjunto de experimentos sobre una segunda secuencia de imágenes en la que se tienen dos robots en movimiento. En la Figura 6.16 se muestran dos imágenes pertenecientes a la secuencia utilizada.

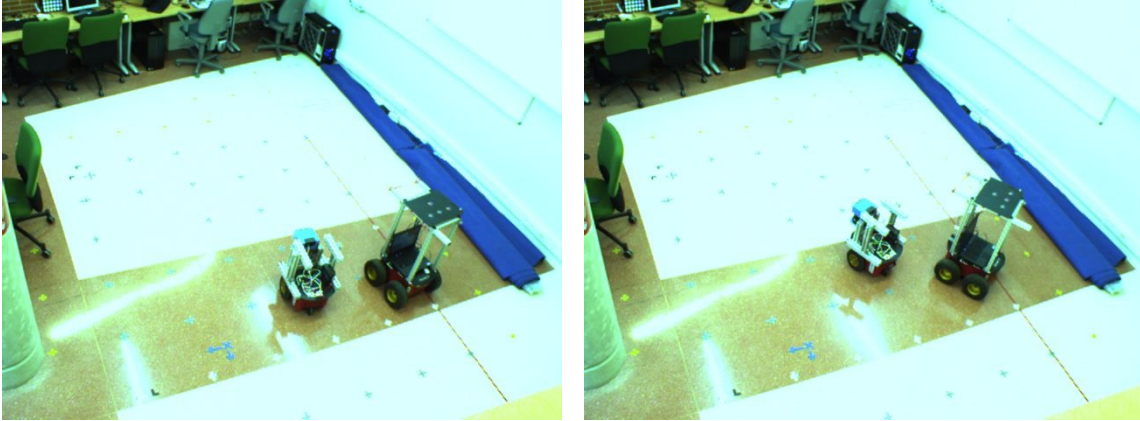


Figura 6.16. Imágenes pertenecientes a la secuencia de prueba.

Estas pruebas experimentales se han llevado a cabo utilizando (6.5) con un valor  $\tau = \frac{2}{3}$ , y el parámetro  $dMax$  se ha fijado utilizando un factor de multiplicación 1.30. En todas las imágenes de la secuencia, el número de robots ha sido estimado de forma correcta. En la Figura 6.17 se muestran los resultados obtenidos para las dos imágenes mostradas en la Figura 6.16.

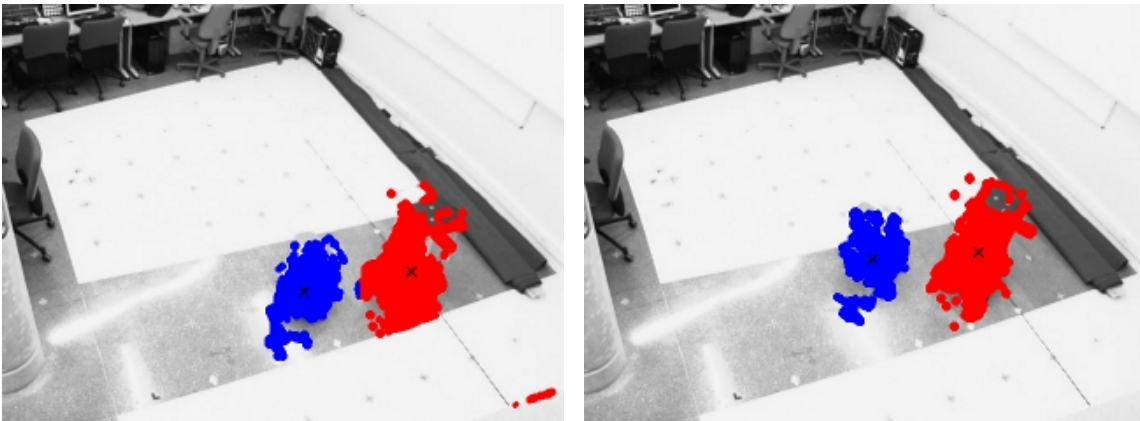


Figura 6.17. Resultado del clustering de las imágenes mostradas en la Figura 6.16 .

Tras las pruebas experimentales realizadas se ha llegado a la conclusión de que el algoritmo de clustering  $k$ -medias extendido permite realizar una estimación del número de robots (y otros objetos) móviles presentes en la escena. Además, proporciona una primera segmentación, que posteriormente se afinará durante la etapa de segmentación de movimiento.

### 6.2.3. Estimación del número de robots móviles en la rejilla de ocupación de *Visual Hull 3D*

Tras el proceso de inicialización de la profundidad usando *Visual Hull 3D* se obtiene una rejilla de ocupación del espacio 3D a partir de la que es posible extraer un conjunto de puntos candidatos a pertenecer a los robots móviles. Sin embargo, de forma previa a la segmentación de movimiento, mediante la minimización de la función objetivo obtenida para  $n_c$  cámaras (y mostrada en la ecuación (4.66)) es necesario estimar el número de robots móviles (u otros posibles obstáculos móviles). Para el caso de una única cámara, esta estimación se realiza utilizando el algoritmo de las  $k$ -medias sobre el plano imagen, según se ha expuesto en el apartado 6.2.2. El algoritmo propuesto para una cámara proporciona resultados adecuados, pero no permite incorporar la información de múltiples cámaras. Por este motivo, en el caso de tener  $n_c$  cámaras se propone utilizar el algoritmo  $k$ -medias extendido, sobre los puntos proyectados al espacio 3D.

Dado que en este caso, el espacio en que se tiene el conjunto de medidas es el espacio cartesiano 3D, parece evidente que los vectores de características deben incluir las tres coordenadas de cada medida en  $\Gamma_w$ . Sin embargo, teniendo en cuenta la aplicación bajo estudio, los robots móviles se desplazan en un plano, la posición de los mismos puede determinarse utilizando únicamente las coordenadas de su centroide en el plano XY. Por este motivo se ha decidido definir el vector de características del algoritmo de clustering en el espacio cartesiano 2D definido por el plano XY del sistema de referencia global ( $\Gamma_w$ ). Esto implica que el proceso de clustering se realizará sobre la proyección del conjunto de medidas 3D en  $\Gamma_w$  sobre el plano XY, de forma que tanto las medidas  $\mathbf{Y}$ , como los centroides de las diferentes clases serán vectores 2D. Respecto a la medida de distancia, en este caso se emplea la distancia euclídea ya que permite conseguir resultados adecuados y su obtención es sencilla.

A modo de ejemplo, en la Figura 6.18(a) se muestra una de las imágenes de las secuencias capturadas en el espacio inteligente. Las medidas aparecen representadas en el sistema de referencia 3D  $\Gamma_w$  en la Figura 6.18(b). También se ha incluido una imagen de la proyección del conjunto de medidas sobre el plano XY. Este conjunto de medidas 2D es el que se empleará como vector de características en el proceso de *clustering*.

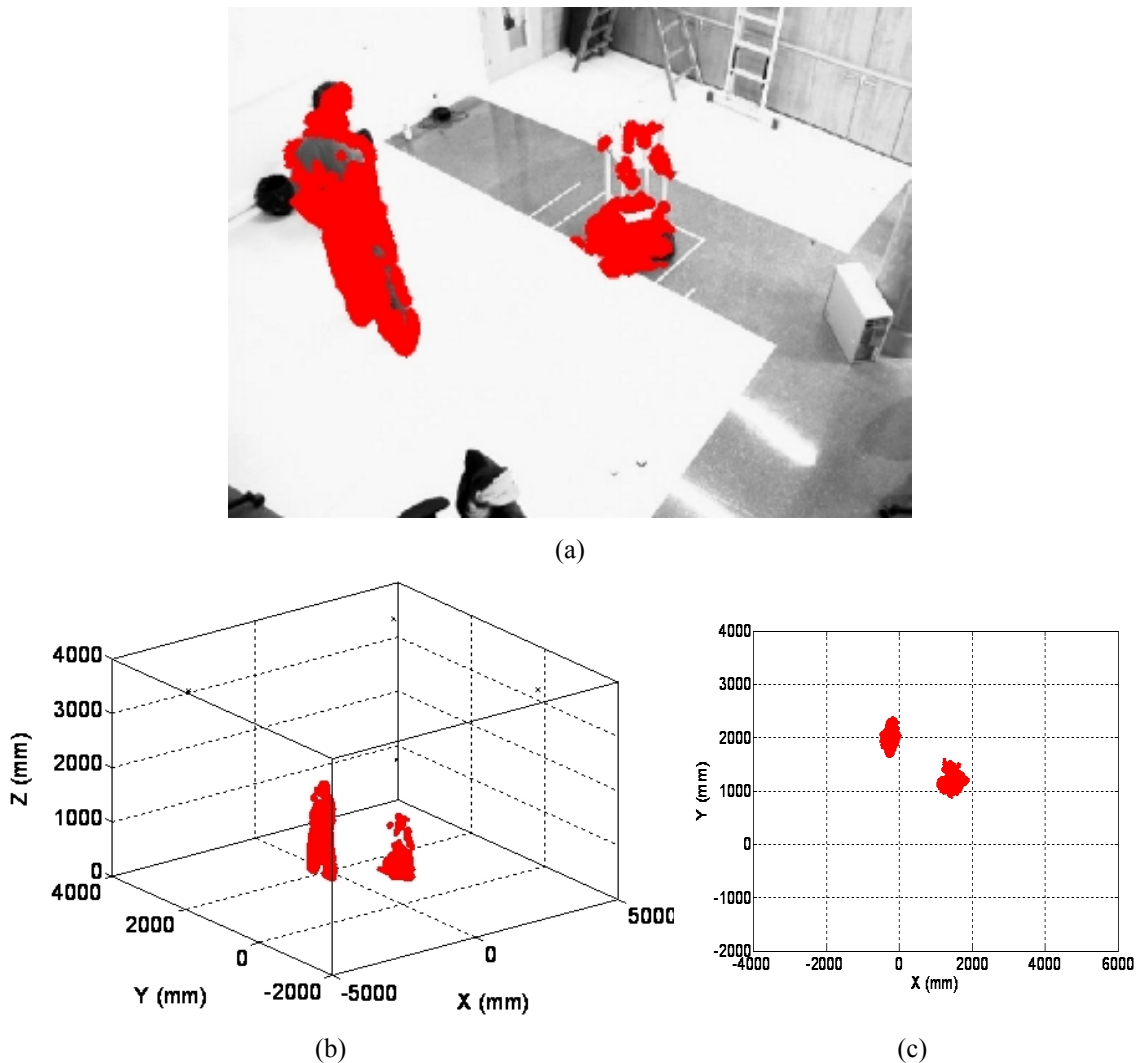


Figura 6.18. Ejemplo de clustering para la detección del número de objetos móviles. (a) Imagen de la escena real captada por una cámara. (b) Conjunto de medidas 3D extraídas de la imagen (a). (c) Proyección sobre el espacio cartesiano 2D definido por el plano XY.

Dado que el algoritmo elegido para el *clustering* es el *k*-medias extendido, es necesario establecer el valor de la distancia límite (que denominamos *dMax*) ya que, como se ha comentado en el apartado 6.2.1.2, los resultados obtenidos dependen fuertemente del valor de este parámetro, pudiendo aparecer clases duplicadas si el valor seleccionado es demasiado pequeño, o clases unidas en caso contrario. Ambos tipos de errores pueden apreciarse en las gráficas de la Figura 6.11(b) y (c), respectivamente.

Para determinar el efecto del valor de *dMax* sobre el resultado del *clustering* se han realizado diversas pruebas experimentales sobre una secuencia de 100 imágenes pertenecientes a la misma secuencia que la imagen mostrada en la Figura 6.18(a). Las medidas obtenidas para cada una de las 100 imágenes se introducen al algoritmo de *k*-medias extendido, considerando diferentes valores del parámetro *dMax*, alrededor de un valor  $d_0=1000$ . Para cada uno de los valores de *dMax*, se ha obtenido el porcentaje de

errores tanto debidos a la duplicación de clases, como a la unión de dos clases. Los resultados se muestran en la Tabla 6.4.

$dMax$	$0.55 \cdot d0$	$0.70 \cdot d0$	$0.85 \cdot d0$	$d0$	$1.15 \cdot d0$	$1.35 \cdot d0$	$1.45 \cdot d0$
<b>Error de duplic. de clases (%)</b>	94	0	0	0	0	0	0
<b>Error de unión de clases (%)</b>	0	0	0	0	100	100	100
<b>% Error total</b>	94	0	0	0	100	100	100

Tabla 6.4. Tasa de error de duplicación de clases y unión de clases obtenidos a partir de una secuencia de 100 imágenes, en función del valor de  $dMax$ .

En esta tabla se observa que, a medida que se incrementa el valor de  $dMax$ , el porcentaje de errores debidos a la duplicación de clases disminuye, mientras que el porcentaje de errores debidos a la unión de dos clases se incrementa. A la vista de los resultados mostrados en la tabla, en el caso de la secuencia de imágenes de prueba considerada, cualquier valor de  $dMax$  comprendido entre 700 y 1000 permite obtener resultados adecuados.

A modo de ejemplo, en la Figura 6.19 se muestra el resultado obtenido tras el *clustering* mediante k-medias extendido con  $dMax = 950$  para un conjunto de imágenes pertenecientes a la misma secuencia que la imagen mostrada en la Figura 6.18(a).

En la columna izquierda de la Figura 6.19 se ha representado el resultado de la clasificación sobre los puntos proyectados al plano  $XY$  en  $\Gamma_w$ , mientras que en la columna derecha se presentan las muestras en 3D en  $\Gamma_w$ .



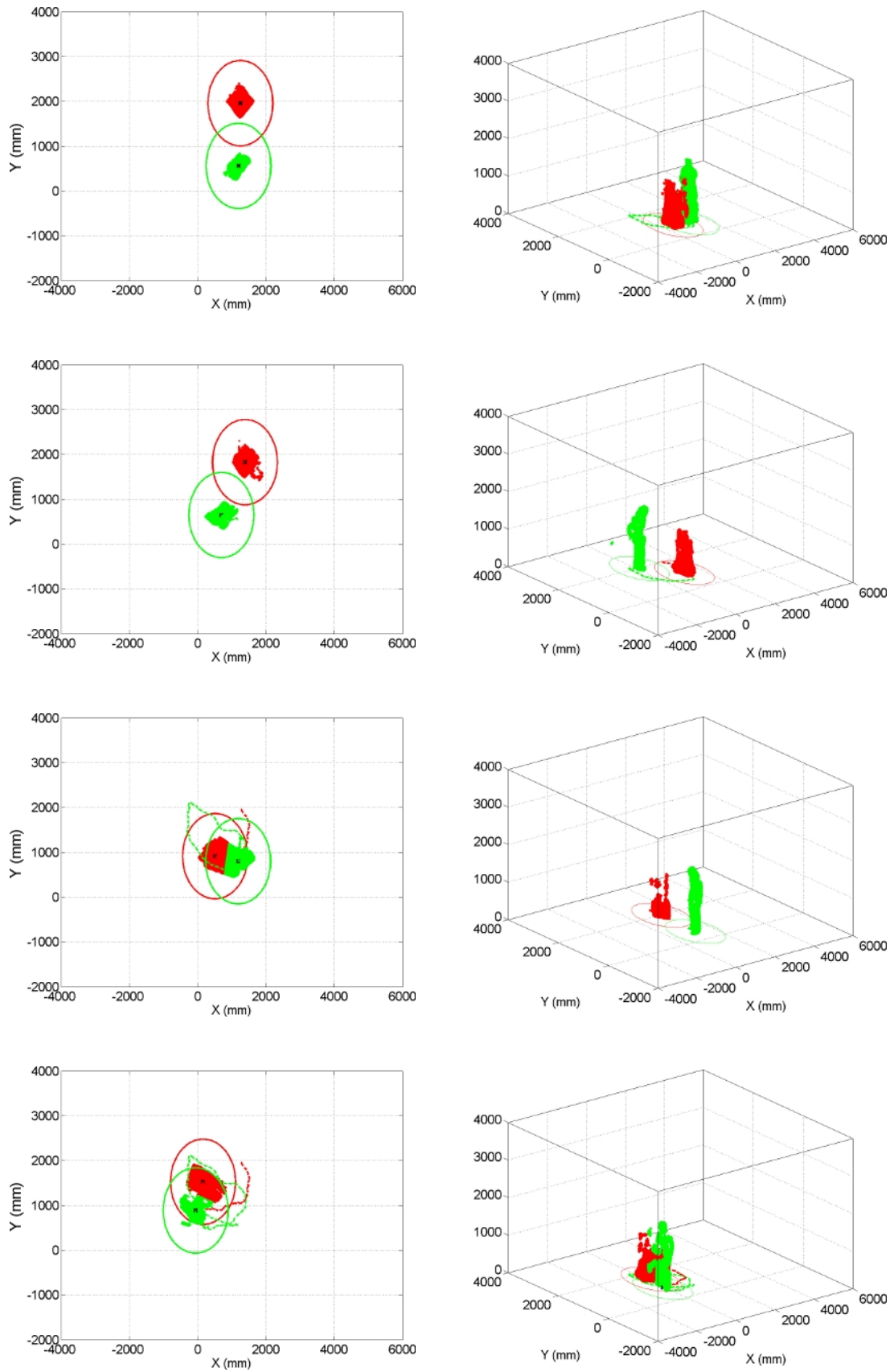


Figura 6.19. Resultado del *clustering*. Las imágenes mostradas en la parte izquierda muestran el resultado de aplicar el algoritmo k-medias sobre el conjunto de muestras proyectadas sobre el plano XY en  $\Gamma_w$ , mientras que las imágenes de la columna derecha representan las muestras en 3D en  $\Gamma_w$ .

### 6.3. Conclusiones y aportaciones

En este capítulo se ha propuesto una solución para la inicialización de la profundidad en aquellos casos en los que se disponga de imágenes de múltiples cámaras. La técnica propuesta está basada en el uso de *Visual Hull 3D* (VH3D) para relacionar la información de las  $n_c$  cámaras disponibles. Esto permite obtener una rejilla de ocupación 3D del espacio, a partir de la cual se obtiene una buena estimación de los valores iniciales de la profundidad.

Por otro lado, la eliminación de las muestras en el sistema de referencia global  $\Gamma_w$  para las que la profundidad es cero, permite reducir las sombras proyectadas por los robots sobre el plano del suelo, sin necesidad de transformar las imágenes a un espacio invariante a la iluminación. Esta alternativa es especialmente útil en los casos en que el número de cámaras que captan simultáneamente los robots sea elevado, ya que proporciona resultados adecuados, y el tiempo de cómputo no tiene una dependencia directa con el número de cámaras.

Asimismo, se han presentado dos alternativas que permiten la estimación del número de robots móviles presentes en la escena. Ambas alternativas están basadas en una versión extendida del algoritmo de *clustering k-medias*. En el primer caso, este algoritmo se aplica sobre los puntos interiores a las curvas sobre el plano imagen. Esta alternativa es especialmente útil para aplicaciones en las que únicamente se dispone de información visual de una cámara. En el caso de tener múltiples cámaras, el uso de *k-medias* extendido para la clasificación del conjunto de puntos 3D, obtenidos tras la inicialización de la profundidad mediante VH3D, proporciona resultados más precisos.

Las aportaciones más destacables presentadas en este capítulo son, por una parte, el uso de VH3D para relacionar la información de todas las cámaras disponibles en el sistema de coordenadas del mundo  $\Gamma_w$ , y por otra, la propuesta para la estimación del número de robots móviles (sobre el plano imagen o en  $\Gamma_w$ ) basada en una versión extendida de *k-medias*.

## Capítulo 7

# **Función de preservación de bordes y ajuste de los valores de las constantes de ponderación de la función objetivo**



Como ya se ha comentado con anterioridad, el objetivo de esta tesis es la obtención de la segmentación de movimiento, y la posición 3D de múltiples robots móviles en un espacio inteligente, a partir de las imágenes adquiridas por múltiples cámaras fijas en el entorno en que se desplazan los robots. Tal como se explica en el capítulo 4, tanto la segmentación, como el posicionamiento se obtienen mediante la minimización de una función objetivo que depende de diferentes grupos de variables relacionadas con los parámetros de movimiento y la posición 3D de dichos robots móviles. Tanto en el caso de una cámara (ecuación (7.1)), como en el de múltiples cámaras (ecuación (7.2)), la función objetivo consta de tres términos diferentes. Los valores de las constantes  $\lambda$  y  $\mu$  ponderan la contribución de cada uno de los términos a la función objetivo, siendo su ajuste de gran importancia debido a la fuerte dependencia de los resultados de los valores de estas constantes.

$$E\left[\{\gamma_k\}_{k=1}^{N-1}, \{\mathbf{v}_{ck}\}_{k=1}^N, \{\boldsymbol{\omega}_{ck}\}_{k=1}^N, Z\right] = \sum_{k=1}^N \left[ \int_{\Omega_k} \psi_k^2(\mathbf{x}) d\mathbf{x} + \mu \int_{\Omega_k} g(\|\nabla Z\|) d\mathbf{x} \right] + \sum_{k=1}^{N-1} \lambda \oint_{\gamma_k} ds \quad (7.1)$$

$$\lambda, \mu \in \mathfrak{R} \text{ y } \lambda, \mu > 0$$

$$E\left[\{\gamma_{ki}\}_{k=1, \dots, N-1}^{i=1, \dots, n_c}, \{\mathbf{T}_{wk}\}_{k=1}^N, \{\boldsymbol{\omega}_{wk}\}_{k=1}^N, \{Z_{ci}\}_{i=1}^{n_c}\right] = \sum_{k=1}^N \sum_{i=1}^{n_{ck}} \left[ \int_{\Omega_{ki}} \psi_{ki}^2(\mathbf{x}) d\mathbf{x} + \mu \int_{\Omega_{ki}} g(\|\nabla Z_{ci}\|) d\mathbf{x} \right] + \sum_{k=1}^{N-1} \sum_{i=1}^{n_{ck}} \lambda \oint_{\gamma_{ki}} ds \quad (7.2)$$

$$\lambda, \mu \in \mathfrak{R} \text{ y } \lambda, \mu > 0$$

En este capítulo se describe el estudio teórico, así como las pruebas experimentales realizadas con el objetivo de elegir los valores más adecuados para las constantes  $\lambda$  y  $\mu$ .

Dado que el segundo término de la función objetivo (que es un término de regularización de la profundidad en ambas funciones) depende de la función de preservación de bordes elegida, de forma previa al ajuste de las constantes se ha realizado un estudio teórico para elegir la función de preservación de bordes más adecuada (de entre las propuestas en [Sekkati & Mítiche 2006b] y [Aubert et al. 1999]). Este estudio, los resultados de los experimentos asociados a él y sus conclusiones se presentan a continuación en el apartado 7.1.

Tras la elección de la función de preservación de bordes, en los apartados 7.2 y 7.3 se describe el proceso de ajuste de los valores de las constantes de ponderación  $\lambda$  y  $\mu$ ,

tanto para el caso de la función objetivo para una única cámara (ecuación (7.1)), como en el caso de múltiples cámaras (ecuación (7.2)).

Cabe destacar que, en las diferentes pruebas experimentales presentadas en este capítulo, para la elección de la función de preservación de bordes y las constantes de ponderación  $\lambda$  y  $\mu$ , se ha empleado una secuencia formada por 500 imágenes (que han sido adquiridas por cuatro cámaras calibradas y sincronizadas entre sí), en las que se desplaza un robot móvil. En la Figura 7.1 se muestran, a modo de ejemplo, tres imágenes pertenecientes a la secuencia de prueba.

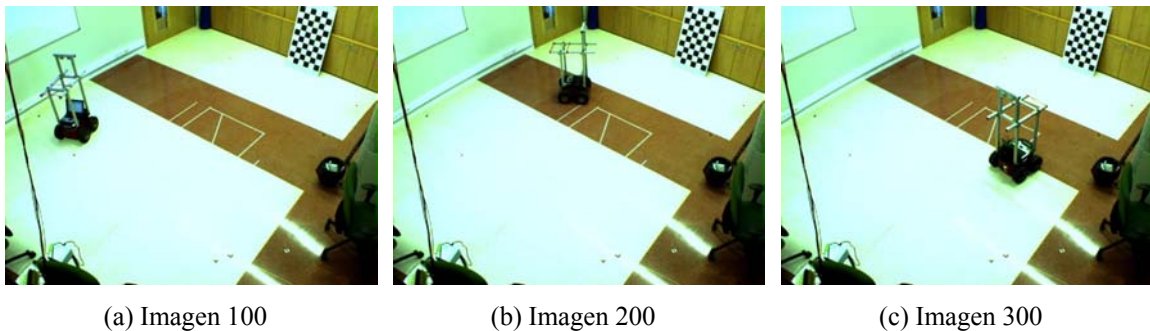


Figura 7.1. Imágenes pertenecientes a la secuencia 1, utilizada para el ajuste de las constantes  $\lambda$  y  $\mu$  y para la elección de la función de preservación de bordes.

Por otro lado, la validación de los valores de  $\lambda$  y  $\mu$  elegidos se ha realizado tanto empleando las imágenes pertenecientes a la secuencia de pruebas mostrada en la Figura 7.1, como utilizando otras dos secuencias de imágenes, adquiridas en el ISPACE-UAH (al igual que la secuencia de la Figura 7.1), pero con diferentes condiciones de iluminación. En la Figura 7.2 se muestra una imagen perteneciente a cada una de estas secuencias adicionales.

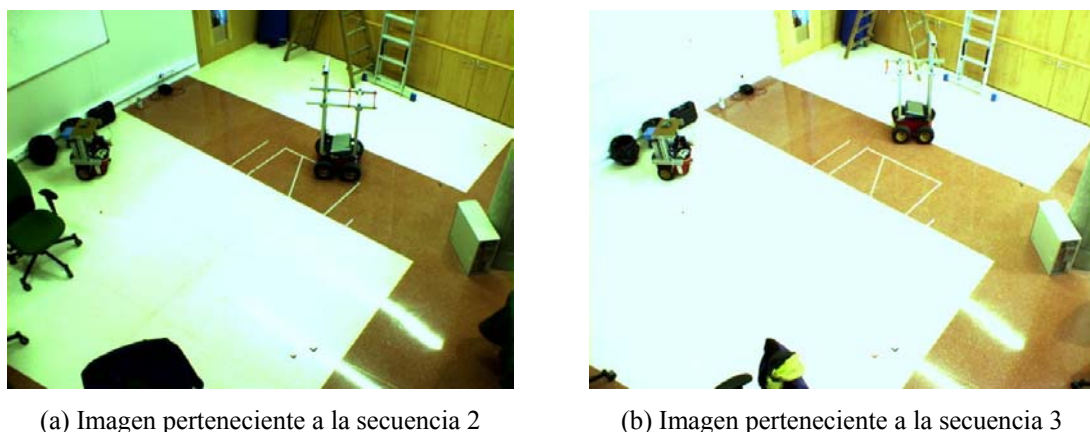


Figura 7.2. Imágenes pertenecientes a la secuencias 2 y 3, utilizadas para validar los valores elegidos para las constantes de ponderación  $\lambda$  y  $\mu$ .

## 7.1. Elección de la función de preservación de bordes

En este apartado se realiza un análisis de las diferentes funciones de preservación de bordes que han sido propuestas en [Sekkti & Mitiche 2006b] y [Aubert et al. 1999]. Además se estudia la relación existente entre las funciones de preservación de bordes y el valor de la constante  $\mu$ . Cabe destacar aquí, que el estudio de la función de preservación de bordes se realiza para la función objetivo definida para una única cámara, si bien los resultados obtenidos en este apartado son fácilmente extrapolables a la función para múltiples cámaras propuesta en esta tesis.

Para comenzar, en la Tabla 7.1 se presentan, a modo de resumen, las funciones de preservación de bordes a analizar, así como sus correspondientes derivadas divididas entre  $\|\nabla Z\|$ .

<b>Función de preservación de bordes (g(s))</b>	$g(\ \nabla Z\ )$	$\frac{g'(\ \nabla Z\ )}{\ \nabla Z\ }$
Cuadrática: $s^2$	$\ \nabla Z\ ^2$	2
Geman-Reynolds: $\frac{s^2}{1+s^2}$	$\frac{\ \nabla Z\ ^2}{1+\ \nabla Z\ ^2}$	$\frac{2}{(1+\ \nabla Z\ ^2)^2}$
Malik-Perona: $\log(1+s^2)$	$\log(1+\ \nabla Z\ ^2)$	$\frac{2}{1+\ \nabla Z\ ^2}$
Green: $2 \cdot \log(\cosh(s))$	$2 \log(\cosh(\ \nabla Z\ ))$	$2 \frac{\tanh(\ \nabla Z\ )}{\ \nabla Z\ }$
Aubert: $2\sqrt{1+s^2} - 2$	$2\sqrt{1+\ \nabla Z\ ^2} - 2$	$\frac{2}{\sqrt{1+\ \nabla Z\ ^2}}$

Tabla 7.1. Resumen de las diferentes funciones para la preservación de bordes propuestas en [Sekkti & Mitiche 2006b] y [Aubert et al. 1999].

A continuación, en la Figura 7.3 se muestran, de forma gráfica, las funciones de preservación de bordes mostradas en la Tabla 7.1, así como sus correspondientes derivadas  $g'(\|\nabla Z\|)/\|\nabla Z\|$  para valores de  $\|\nabla Z\|$  entre 0 y +2. Sólo se ha representado la parte positiva de la función debido a que la  $\|\nabla Z\|$  nunca toma valores negativos. En la Figura 7.3(a) se observa que todas las funciones de preservación de bordes bajo estudio son funciones crecientes, y de valor nulo cuando el valor de  $\|\nabla Z\|$  es igual a cero. Respecto a las derivadas mostradas en la Figura 7.3(b), todas ellas tienen su valor máximo (2) cuando  $\|\nabla Z\|$  es cero y decrecen a medida que aumenta el valor de  $\|\nabla Z\|$ .

Estas curvas permiten evaluar cuál es el efecto de utilizar una u otra función de restricción de bordes en la función objetivo ya que, cuanto mayor sea la pendiente con la que decrece la curva de la derivada, menor será el valor que tome el segundo término de la ecuación de descenso por el gradiente. De forma que, para un valor de la constante  $\mu$  fijo, el protagonismo del término asociado a la norma del gradiente de la profundidad en la ecuación de descenso será mayor para aquellas funciones de preservación de bordes cuya derivada presente una menor pendiente.

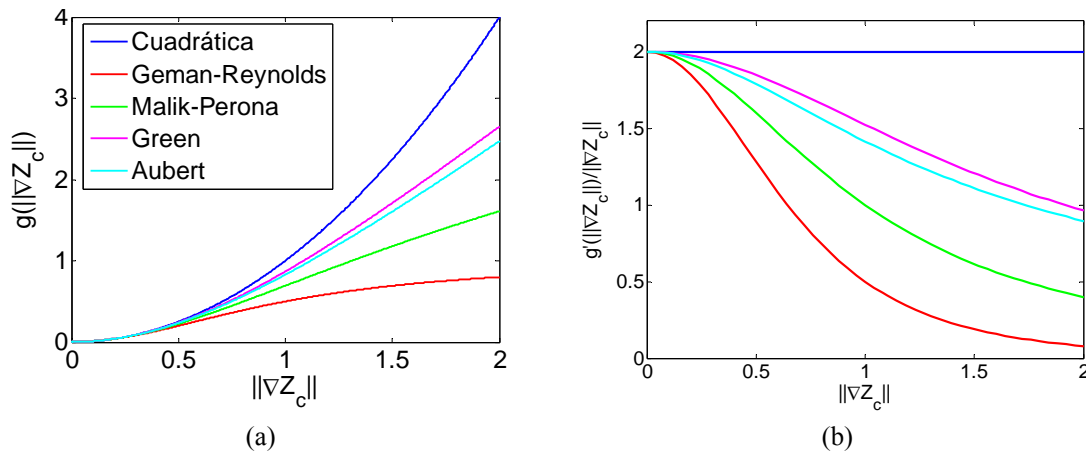


Figura 7.3. (a) Representación gráfica de las funciones de preservación de bordes presentadas en la Tabla 7.1 para valores de  $\|\nabla Z\|$  comprendidos entre 0 y 2. (b) Representación gráfica de las derivadas de las funciones de preservación de bordes divididas entre  $\|\nabla Z\|$ .

Con objeto de evaluar el efecto sobre la ecuación de descenso por el gradiente del término dependiente del gradiente de la profundidad, para cada una de las funciones de preservación de bordes contempladas en la Tabla 7.1, se ha calculado el valor máximo y mínimo que toma el citado término para 100 imágenes de prueba algunas de las cuales se muestran a modo de ejemplo en la Figura 7.1. En la Tabla 7.2 se muestran los valores medios obtenidos para las 100 imágenes consideradas, así como el tiempo de cómputo consumido para cada una de las funciones de preservación de bordes bajo estudio.

Al igual que en la Figura 7.3(b), en la Tabla 7.2 se observa que existen grandes diferencias en el rango de valores que toma el segundo término de la ecuación de descenso por el gradiente, para las diferentes funciones de preservación de bordes consideradas. Además, se puede observar también una relación entre la pendiente de las curvas mostradas en la Figura 7.3(b) y los valores presentados en la Tabla 7.2, de forma que, cuanto mayor es la pendiente de las curvas, menores son los valores que toma dicho término.



Por otro lado, el valor de la constante  $\mu$  está directamente relacionado con los valores que toma el segundo término de la ecuación de descenso por el gradiente, ya que pondera la contribución de ese término, frente a la restricción de brillo 3D.

Función de preservación de bordes (g(s))	Valor máximo	Valor mínimo	Tiempo de cómputo (s)
Cuadrática: $s^2$	524.4635	-656.8217	0.2524
Geman-Reynolds: $\frac{s^2}{1+s^2}$	0.0022	-0.0016	0.2811
Malik-Perona: $\log(1+s^2)$	0.0146	-0.0178	0.2794
Green: $2 \cdot \log(\cosh(s))$	1.2106	-1.6088	0.2693
Aubert: $2\sqrt{1+s^2} - 2$	1.2099	-1.6080	0.3575

Tabla 7.2. Rango de valores y tiempo de cómputo consumido en el cálculo del término dependiente del gradiente de la profundidad en la ecuación de descenso por el gradiente, para las diferentes funciones de preservación de bordes (g(s)) consideradas. Valores medios obtenidos para 100 imágenes de prueba capturadas por 4 cámaras.

Con objeto de determinar el efecto que tiene cada una de las funciones sobre el resultado de la segmentación de movimiento, se ha ajustado el valor de la constante  $\mu$ . El objetivo del ajuste es que el segundo término de la ecuación de descenso por el gradiente tome valores en el mismo orden de magnitud, independientemente de la función de preservación de bordes utilizada.

Para las pruebas iniciales se ha decidido ajustar el valor de la constante  $\mu$  de forma que el orden de magnitud del segundo término de la ecuación de descenso por el gradiente de la profundidad sea unas 100 veces mayor que el primer término. Más adelante, en este mismo apartado, se presentarán las pruebas realizadas para el ajuste del valor de la constante  $\mu$ . Dado que para las imágenes disponibles se ha comprobado que el primer término de la ecuación de descenso toma valores del orden de  $10^{-4}$ , el valor de  $\mu$  para esta prueba se ha fijado de forma que el valor del segundo término se encuentre en el orden de magnitud de  $10^{-2}$ . En la segunda columna de la Tabla 7.3 se muestra el orden de magnitud de los valores de  $div\left(\frac{g(\|\nabla Z\|)}{\|\nabla Z\|} \nabla Z\right)$  para las imágenes de prueba consideradas, mientras que en la tercera columna de esta tabla se encuentran los valores de  $\mu$  que se han elegido para cada una de las funciones de preservación de bordes.

<b>Función de preservación de bordes</b>	<b>Orden de magnitud de <math>div\left(\frac{g(\ \nabla Z\ )}{\ \nabla Z\ }\nabla Z\right)</math></b>	<b>Orden de magnitud de la constante <math>\mu</math></b>
<b>Cuadrática</b>	$10^2$	$10^{-4}$
<b>Geman-Reynolds</b>	$10^{-3}$	$10^1$
<b>Malik-Perona</b>	$10^{-2}$	$10^0$
<b>Green</b>	$10^0$	$10^{-2}$
<b>Aubert</b>	$10^0$	$10^{-2}$

Tabla 7.3. Orden de magnitud de los valores del segundo término de la función objetivo (ecuación (7.1)) para las imágenes de prueba y de la constante  $\mu$ , para que los dos términos de dicha ecuación tomen valores en el mismo orden de magnitud.

Para determinar la validez de esta propuesta se ha realizado un conjunto de pruebas experimentales sobre 100 imágenes en las que aparece un robot móvil. Para estas pruebas el valor de la constante  $\lambda$  se ha fijado en 20.000. Este valor de  $\lambda$  se ha elegido con objeto de independizar el efecto de la constante  $\mu$  y la función de preservación de bordes del efecto de la constante  $\lambda$ . Esto se hace así porque al utilizar un valor de  $\lambda$  elevado, las curvas apenas evolucionan en cada iteración (se da un peso mucho mayor al término asociado a la curvatura en la iteración anterior) de forma que los cambios en las diferentes variables se deben al efecto de la  $\mu$  y la función de preservación de bordes elegidas.

En las pruebas experimentales realizadas se ha puesto de manifiesto que al utilizar los valores de la constante  $\mu$  mostrados en la Tabla 7.3, tanto el número de iteraciones necesarias hasta alcanzar la convergencia del algoritmo, como los resultados obtenidos (velocidad lineal y angular 3D, profundidad y contorno de la segmentación) son muy similares para todos los casos. Esto implica que cualquiera de las funciones de preservación de bordes presentadas en la Tabla 7.1 es válida para este trabajo.

En la Figura 7.4 se representa el tiempo de cómputo empleado por el proceso de minimización de energía de cada una de las 100 imágenes de entrada. El tiempo consumido por la función de Aubert es superior al que requiere cualquiera de las otras funciones. Respecto al resto de funciones, todas ellas presentan requerimientos de tiempo similares (motivo por el que aparecen prácticamente superpuestas en la Figura 7.4), siendo ligeramente menores los tiempos consumidos por la función cuadrática.

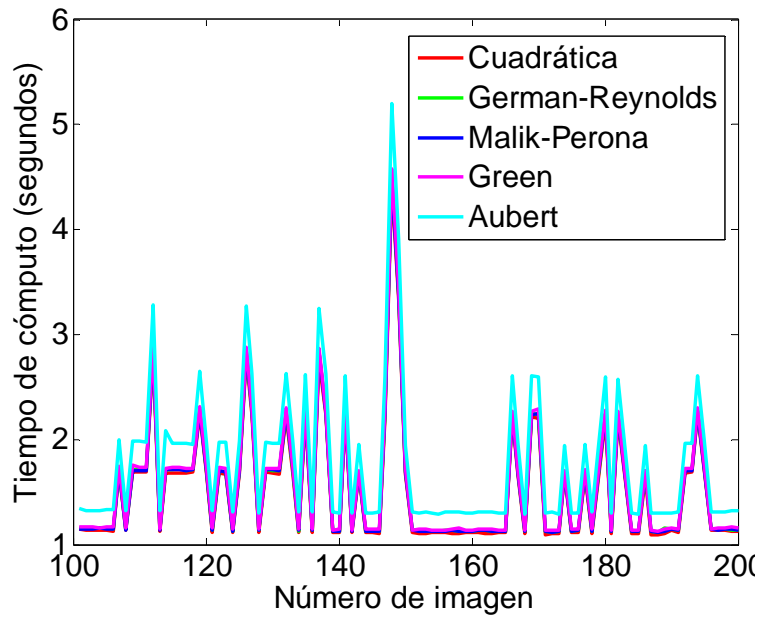


Figura 7.4. Tiempo de cómputo consumido por la minimización de la función objetivo calculado para 100 imágenes, para las diferentes funciones de preservación de bordes consideradas.

Tras el estudio realizado en este apartado se ha decidido utilizar la función de bordes cuadrática ya que, a pesar de ser la más sencilla, proporciona resultados adecuados, siendo además su tiempo de cómputo ligeramente inferior al del resto de funciones estudiadas.

Una vez elegida la función de preservación de bordes a utilizar, es necesario ajustar el valor de la constante  $\mu$  para esta función, de forma que se obtengan resultados precisos en el mínimo número de iteraciones del algoritmo. Una vez obtenido el valor de  $\mu$  más adecuado para la función cuadrática, es sencillo extrapolar este resultado para el resto de funciones, multiplicando por un factor de escala.

Cabe destacar aquí que, aunque los resultados presentados en este apartado han sido obtenidos para la función objetivo propuesta en [Sekkti & Mitiche 2006b], estos resultados pueden extrapolarse, de forma sencilla, para la función objetivo propuesta en este trabajo (ecuación (7.2)) en la que se incorpora la información adquirida por múltiples cámaras en el espacio inteligente.

## **7.2. Valores de las constantes de ponderación $\lambda$ y $\mu$ para la función objetivo definida para una cámara**

Tras la elección de la función de preservación de bordes, en este apartado se presenta el estudio realizado para el ajuste de las constantes  $\lambda$  y  $\mu$  que ponderan la contribución de cada uno de los términos a la función objetivo definida en [Sekkati & Mitiche 2006b].

### **7.2.1. Ajuste de la constante $\mu$**

Para la obtención del valor de  $\mu$  se ha realizado un conjunto de pruebas experimentales. En dichas pruebas se han obtenido los resultados de la segmentación de movimiento para una secuencia de 100 imágenes en las que aparece un robot móvil para diferentes valores de la constante  $\mu$ . En concreto, los valores de  $\mu$  considerados para las pruebas experimentales realizadas han sido:  $10^{-3}$ ,  $5 \cdot 10^{-3}$ ,  $10^{-2}$ ,  $5 \cdot 10^{-2}$ ,  $10^{-1}$ ,  $5 \cdot 10^{-1}$ , 1. No se han incluido resultados para valores de  $\mu$  superiores a 1 debido a que se ha comprobado que para dichos valores, al utilizar la función cuadrática de preservación de bordes, el proceso de minimización de energía no alcanza la convergencia.

En cada una de estas pruebas se ha evaluado la calidad de la segmentación resultante, así como el error cometido en el cálculo de las velocidades. Este error se ha calculado en todos los casos como la diferencia entre las velocidades estimadas por los sensores odométricos del robot, y las obtenidas tras la minimización de la función objetivo. Además, también se ha tenido en cuenta el número de iteraciones necesarias hasta la convergencia del algoritmo.

Para la realización de todas estas pruebas se ha considerado un valor de 20.000 para la constante  $\lambda$  ya que al inicializar las curvas en posiciones cercanas a los contornos de los robots móviles presentes no es necesario que se produzcan grandes cambios en las curvas durante el proceso de minimización de energía. Además, con objeto de que la evolución de las curvas sea lenta, se fija  $\lambda$  a un valor elevado para tratar de independizar el efecto de la constante  $\mu$  del de la constante  $\lambda$ .

### 7.2.1.1. Efecto del valor de la constante $\mu$ sobre las curvas que definen los contornos de las regiones en que se segmenta la imagen

En las pruebas experimentales realizadas, con los diferentes valores de la constante  $\mu$  considerados, se ha puesto de manifiesto que la influencia de dicha constante sobre las curvas que definen los contornos de las diferentes regiones en que se divide la imagen es insignificante.

En la Figura 7.5(a) se muestra la evolución del número de píxeles contenidos por la curva que define el contorno del robot móvil para 10 imágenes de la secuencia de prueba a la que pertenecen las imágenes de la Figura 7.1. En esta figura las curvas aparecen superpuestas ya que en todos los casos la variación del número de píxeles en función de  $\mu$  es muy pequeña. Esto también se aprecia en la Figura 7.5(b) en la que se han representado las curvas que definen el contorno del robot móvil para la imagen 221 de la secuencia de imágenes de prueba. Los resultados mostrados en la Figura 7.5(a) y (b) demuestran que el valor elegido para la constante  $\mu$  no afecta a las curvas iniciales obtenidas. Esto se debe en gran medida al valor de  $\lambda$  elegido, ya que, aunque a medida que se incrementa el valor de  $\mu$  se ejecuta un número mayor de iteraciones hasta la convergencia, al elegir para  $\lambda$  un valor elevado, las curvas apenas cambian en cada iteración.

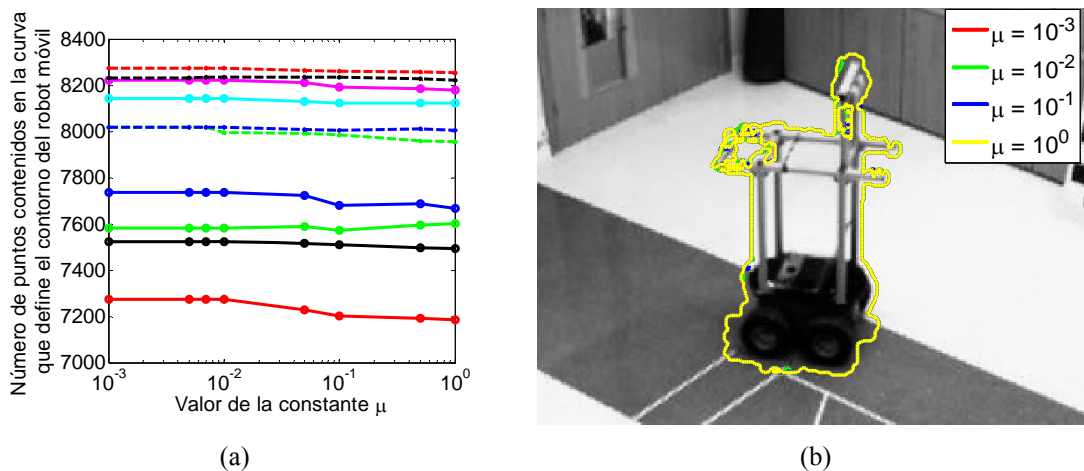


Figura 7.5. (a) Número de píxeles contenidos en la curva que define el contorno del robot móvil en función del valor de la constante  $\mu$  para 10 imágenes de prueba de la secuencia a la que pertenecen las imágenes de la Figura 7.1. (b) Detalle de las curvas obtenidas para una imagen de la secuencia.

### 7.2.1.2. Efecto del valor de la constante $\mu$ sobre la profundidad.

El efecto de la constante  $\mu$  sobre la profundidad en cada píxel de la imagen es más significativo que en el caso de las curvas. En la Figura 7.6 se muestra, a modo de ejemplo, el valor de la profundidad en algunos de los píxeles de la imagen.

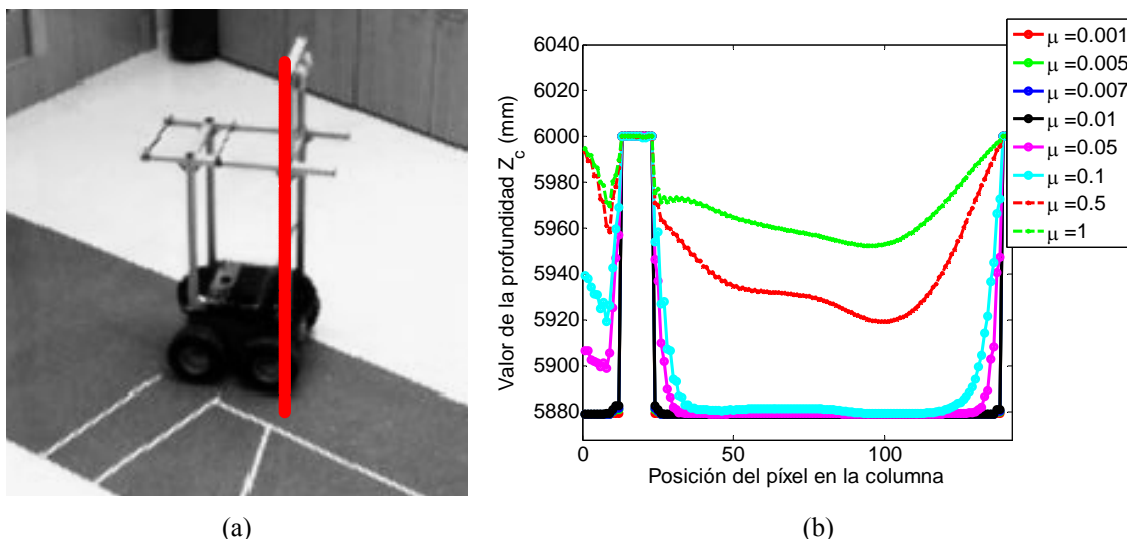


Figura 7.6. (a) Detalle de la imagen de entrada sobre la que se han resaltado en color rojo los píxeles para los que se representa el valor de la profundidad. (b) Valor de la profundidad en los píxeles marcados en rojo en (a) en función del valor de la constante  $\mu$ .

En esta figura se puede observar como, a medida que aumenta el valor de  $\mu$ , la profundidad en los píxeles del robot toma valores más elevados. Sin embargo, en la región de la imagen para la que se ha realizado la representación, la diferencia entre las profundidades obtenidas con  $\mu=10^{-3}$ , y con  $\mu=1$  no supera en ningún caso los 120 mm, siendo este valor pequeño en comparación con la distancia a la que se encuentra el robot de la cámara, que es del orden de los 4 metros.

Con objeto de evaluar las diferencias en la profundidad en función del valor de  $\mu$ , en la Figura 7.7 se representan los valores máximos de las diferencias entre la profundidad estimada con  $\mu=10^{-3}$ , y valores de  $\mu= 5 \cdot 10^{-3}, 10^{-2}, 5 \cdot 10^{-2}, 10^{-1}, 5 \cdot 10^{-1}, 10^0$ .

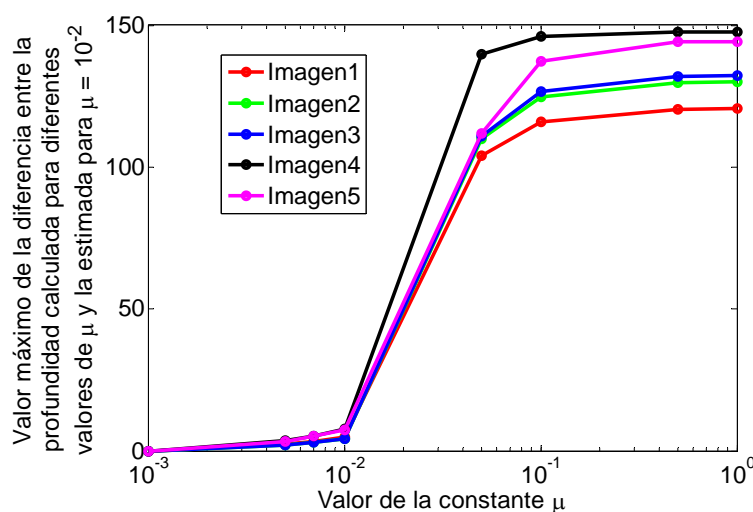


Figura 7.7. Valor máximo de la diferencia entre la profundidad obtenida para  $\mu = 10^{-3}$  y para los siguientes valores de  $\mu$ :  $5 \cdot 10^{-3}, 10^{-2}, 5 \cdot 10^{-2}, 10^{-1}, 5 \cdot 10^{-1}, 10^0$ .

En la Figura 7.7 se puede observar que a medida que se incrementa el valor de  $\mu$  se incrementa el valor de la profundidad obtenida. Sin embargo, en el peor de los casos, esta diferencia no supera los 150 mm para todos los valores de  $\mu$  considerados. Dado que esta diferencia de distancias es pequeña comparándola con la distancia del robot a la cámara (más de 4 metros) la variación de la profundidad  $Z_c$  no es decisiva para la elección del valor de la constante  $\mu$ .

### 7.2.1.3. Efecto del valor de la constante $\mu$ sobre las componentes de la velocidad 3D en el sistema de referencia de la cámara

Finalmente se analiza el efecto del valor de la constante  $\mu$  sobre las componentes de velocidad 3D. Para ello se comparan los resultados obtenidos tras el proceso de minimización de la función objetivo con las medidas de los sensores odométricos del robot. Esta comparación se realiza en el sistema de referencia global  $\Gamma_w$  debido a que las medidas del robot se encuentran representadas en ese sistema de referencia. Por tanto, es necesario proyectar las velocidades obtenidas en  $\Gamma_c$ , al sistema de referencia  $\Gamma_w$  utilizando para ello las matrices de calibración de la cámara.

Siendo  $\mathbf{P}_c = (X_c, Y_c, Z_c)^T$  las coordenadas 3D de un punto  $\mathbf{P}$  perteneciente al robot móvil en el sistema de referencia de la cámara  $\Gamma_c$  y  $\mathbf{P}_w = (X_w, Y_w, Z_w)^T$  las coordenadas 3D de dicho punto en el sistema de referencia global  $\Gamma_w$ , la velocidad de  $\mathbf{P}$  en  $\Gamma_c$  y  $\Gamma_w$  puede expresarse (ver (7.3) y (7.4)) como:

$$\dot{\mathbf{P}}_c = \mathbf{v}_c + \boldsymbol{\omega}_c \times \mathbf{P}_c = \mathbf{v}_c + \hat{\boldsymbol{\omega}}_c \cdot \mathbf{P}_c \quad (7.3)$$

$$\dot{\mathbf{P}}_w = \mathbf{v}_w + \boldsymbol{\omega}_w \times \mathbf{P}_w = \mathbf{v}_w + \hat{\boldsymbol{\omega}}_w \cdot \mathbf{P}_w \quad (7.4)$$

en las que  $\hat{\boldsymbol{\omega}}$  es la matriz antisimétrica.

$$\hat{\boldsymbol{\omega}} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$$

Por otro lado, conocidas las matrices de transformación entre el sistema de referencia global y el de la cámara ( $\mathbf{R}_{wc}$  y  $\mathbf{T}_{wc}$ ), se cumple:

$$\mathbf{P}_c = \mathbf{R}_{wc} \cdot \mathbf{P}_w + \mathbf{T}_{wc} \quad (7.5)$$

Si se deriva la ecuación (7.5) se obtiene la relación entre la velocidad del punto  $\mathbf{P}$  en  $\Gamma_c$  y  $\Gamma_w$ .

$$\dot{\mathbf{P}}_c = \mathbf{R}_{wc} \cdot \dot{\mathbf{P}}_w \quad (7.6)$$

Sustituyendo las expresiones de  $\dot{\mathbf{P}}_c$ , (7.3),  $\dot{\mathbf{P}}_w$ , (7.4), y  $\mathbf{P}_c$  (7.5) en (7.6), se obtiene:

$$\mathbf{v}_c + \hat{\boldsymbol{\omega}}_c (\mathbf{R}_{wc} \mathbf{P}_w + \mathbf{T}_{wc}) = \mathbf{R}_{wc} \mathbf{v}_w + \mathbf{R}_{wc} \hat{\boldsymbol{\omega}}_w \mathbf{P}_w \quad (7.7)$$

A partir de (7.7) se obtiene la expresión de las componentes de velocidad lineal y angular en  $\Gamma_w$ , en función de las velocidades 3D en  $\Gamma_c$ .

$$\mathbf{v}_c + \hat{\boldsymbol{\omega}}_c \mathbf{T}_{wc} = \mathbf{R}_{wc} \mathbf{v}_w \Rightarrow \mathbf{v}_w = \mathbf{R}_{wc}^{-1} (\mathbf{v}_c + \hat{\boldsymbol{\omega}}_c \mathbf{T}_{wc}) \quad (7.8)$$

$$\hat{\boldsymbol{\omega}}_c \mathbf{R}_{wc} = \mathbf{R}_{wc} \hat{\boldsymbol{\omega}}_w \Rightarrow \hat{\boldsymbol{\omega}}_w = \mathbf{R}_{wc}^{-1} \hat{\boldsymbol{\omega}}_c \mathbf{R}_{wc} \quad (7.9)$$

Las ecuaciones (7.8) y (7.9) permiten transformar las velocidades calculadas mediante la minimización de la función objetivo del sistema de referencia  $\Gamma_c$  al sistema de referencia global  $\Gamma_w$ , para posteriormente poder comparar estas velocidades con las medidas de los sensores odométricos del robot.

Para evaluar el efecto de la constante  $\mu$  sobre las velocidades, se ha obtenido el error en la velocidad lineal y angular, calculado como la diferencia entre el módulo de la velocidad obtenida usando el algoritmo propuesto, y la medida por los sensores odométricos del robot, para un conjunto de 100 imágenes de prueba de la secuencia a la que pertenecen las imágenes de la Figura 7.1.

En la Figura 7.8 se muestra el error cuadrático medio de la velocidad lineal (en milímetros/segundo) calculado para 5 imágenes (no consecutivas) de la secuencia en  $\Gamma_w$ . Este error se calcula utilizando (7.10). En caso de que el robot se mueva por un plano, esta ecuación se reduce, debido a que la componente en  $Z$  ( $v_{wz}$ ) es nula en  $\Gamma_w$ .

$$\varepsilon_{v_w} = \sqrt{v_{wx}^2 + v_{wy}^2 + v_{wz}^2} \quad (7.10)$$

En la Figura 7.8(a) se puede observar que para valores de  $\mu$  próximos a  $10^{-1}$ , el error en la velocidad lineal tiene un mínimo para todas las imágenes incluidas en la gráfica. Por otro lado, en la Figura 7.8(b) se ha representado el error cuadrático medio en la velocidad angular (en grados/segundo) para las mismas imágenes. En este caso se puede observar como las gráficas del error apenas varían para los diferentes valores de la constante  $\mu$  considerados.



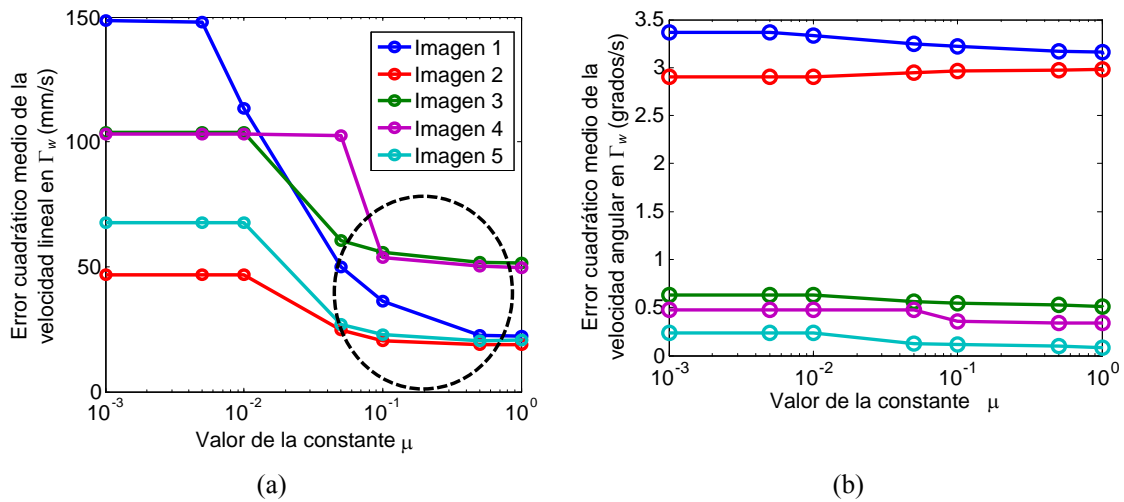


Figura 7.8. Errores cometidos en el cálculo de las velocidades lineal y angular calculados como la diferencia entre las velocidades estimadas por el algoritmo, y las medidas por los sensores odométricos del robot en  $\Gamma_w$ . (a) Error cometido en el cálculo de la velocidad lineal. (b) Error cometido en el cálculo de la velocidad angular.

A continuación, en la Tabla 7.4 se muestra el valor del error cuadrático medio de la velocidad lineal para las 100 imágenes de la secuencia de prueba, así como el número de iteraciones necesarias hasta la convergencia. En esta tabla, al igual que en la Figura 7.8, se puede observar que el error cuadrático medio cometido en la velocidad lineal presenta su valor mínimo cuando la constante  $\mu$  toma valores en torno a  $10^{-1}$ . Para valores de  $\mu$  superiores a 1 se presentan problemas de convergencia.

Valor de la constante $\mu$	Número medio de iteraciones	Valor medio del error en la velocidad lineal (mm/s)
$10^{-3}$	2.48	89.3544
$5 \cdot 10^{-3}$	3.68	88.6540
$10^{-2}$	21.68	77.5520
$5 \cdot 10^{-2}$	41.88	43.2021
$10^{-1}$	38.48	31.0856
$5 \cdot 10^{-1}$	19.00	31.8055
1	13.84	32.8893
5	No converge	-

Tabla 7.4. Valor medio del error cometido en la estimación de la velocidad lineal y número medio de iteraciones necesarias hasta la convergencia del algoritmo para las 100 imágenes de la secuencia de prueba en función del valor de la constante  $\mu$ , para valores de  $\mu$  entre  $10^{-3}$  y 1.

Observando los resultados presentados en la Tabla 7.4, se comprueba que, para valores de la constante  $\mu$  inferiores a  $10^{-1}$ , a medida que se incrementa el valor de la constante  $\mu$ , aumenta el número de iteraciones hasta la convergencia, y disminuye el valor del error cuadrático medio de la velocidad lineal. Sin embargo, a partir de  $10^{-1}$ , el

efecto observado es el opuesto, disminuyendo el número de iteraciones, y aumentando el error en la velocidad.

Partiendo de los resultados mostrados en la Tabla 7.4, se ha realizado un nuevo análisis para valores de  $\mu$  entre  $5 \cdot 10^{-2}$  y 1, con objeto de determinar en qué rango de valores de  $\mu$  se obtiene una mejor relación entre el número de iteraciones y el error cuadrático medio en la velocidad lineal, para las 100 imágenes de prueba de la secuencia a la que pertenecen las imágenes de la Figura 7.1. Los resultados obtenidos se presentan en la Tabla 7.5.

Valor de la constante $\mu$	Número medio de iteraciones	Error cuadrático medio de la velocidad lineal (mm/s)
$5 \cdot 10^{-2}$	41.88	43.2021
$7 \cdot 10^{-2}$	43.76	32.8853
$10^{-1}$	38.48	31.0856
$3 \cdot 10^{-1}$	23.92	30.9616
$5 \cdot 10^{-1}$	19.00	31.8055
$7 \cdot 10^{-1}$	16.28	32.3164
1	13.84	32.8893

Tabla 7.5. Valor medio del error cometido en la estimación de la velocidad lineal y número medio de iteraciones necesarias hasta la convergencia del algoritmo, para las 100 imágenes de la secuencia de prueba para valores de  $\mu$  entre  $5 \cdot 10^{-2}$  y 1.

Como ya se ha comentado, es necesario alcanzar una solución de compromiso entre el número de iteraciones y el error cometido en el cálculo de la velocidad. En el caso de las imágenes de prueba, se observa que el error cuadrático medio de la velocidad lineal presenta su valor mínimo para  $\mu=3 \cdot 10^{-1}$ , mientras que el mínimo número de iteraciones hasta la convergencia se produce para  $\mu=1$ . En la Tabla 7.5 se han señalado en color verde los valores para los que se existe una mejor relación entre el número de iteraciones y el error en la velocidad. En concreto, los valores de  $\mu$  elegidos son  $3 \cdot 10^{-1}$  y  $5 \cdot 10^{-1}$ . En cualquier caso, para las imágenes de prueba, el error cuadrático medio en la velocidad lineal es inferior a 40 mm/s siempre que el valor de  $\mu$  sea superior a  $5 \cdot 10^{-2}$ . A la vista de los resultados presentados, para las imágenes pertenecientes a la secuencia 1 (Figura 7.1) el valor de  $\mu$  elegido debe encontrarse entre  $7 \cdot 10^{-2}$  y 1. Estos resultados han sido validados para otras secuencias adquiridas en el ISPACE-UAH, y serán mostrados más adelante en este mismo capítulo.

## 7.2.2. Ajuste de la constante $\lambda$

Para el ajuste del valor de la constante  $\lambda$  para las imágenes de prueba se han realizado un conjunto de pruebas experimentales en las que se han evaluado los resultados de la segmentación de movimiento y el posicionamiento 3D de un robot móvil, a partir de 100 imágenes captadas en el espacio inteligente. Algunas de las imágenes de la secuencia utilizada se muestran en la Figura 7.1.

Al igual que en el caso del ajuste de la constante  $\mu$ , en cada una de las pruebas experimentales realizadas se ha evaluado tanto la calidad de la segmentación resultante, como el error cometido en el cálculo de las velocidades (calculado como la diferencia entre las velocidades estimadas por los sensores odométricos del robot, y las obtenidas tras la minimización de la función objetivo). Además, también se ha tenido en cuenta el número de iteraciones necesarias hasta la convergencia del algoritmo. En todas las pruebas realizadas, el valor de la constante  $\mu$  se ha fijado a  $5 \cdot 10^{-1}$ .

A continuación se evalúa el efecto de la constante  $\lambda$  sobre cada uno de los grupos de variables implicados en la función objetivo definida en la ecuación (7.1): las curvas que definen los contornos de los robots móviles, la profundidad y las componentes de velocidad 3D.

### 7.2.2.1. Efecto del valor de la constante $\lambda$ sobre la profundidad

En primer lugar se evaluó el efecto de la constante  $\lambda$  sobre la profundidad en cada píxel de la imagen. En las pruebas realizadas se ha puesto de manifiesto que la constante  $\lambda$  apenas influye sobre la profundidad. Esto se debe a que no está incluida en la ecuación de descenso por el gradiente, que es la que determina la evolución de la profundidad en cada iteración. Sin embargo, se ha considerado necesario estudiar el efecto de  $\lambda$  sobre la profundidad, debido a que su valor determina la variación de los contornos de la segmentación en cada iteración y las curvas que definen esos contornos sí se incluyen en la ecuación de descenso por el gradiente para obtener la profundidad en cada iteración.

En la Figura 7.9(b) se ha representado el valor de la profundidad en los píxeles marcados en rojo en la imagen de la Figura 7.9(a) para diferentes valores de la constante  $\lambda$  entre 2 y 20000. En esta gráfica se observa que las curvas de la profundidad para distintos valores de  $\lambda$  aparecen superpuestas, es decir, la profundidad apenas se

modifica al variar la constante  $\lambda$ . Los pequeños cambios que se aprecian se deben a la diferencia en el número de iteraciones que se ejecutan en cada caso.

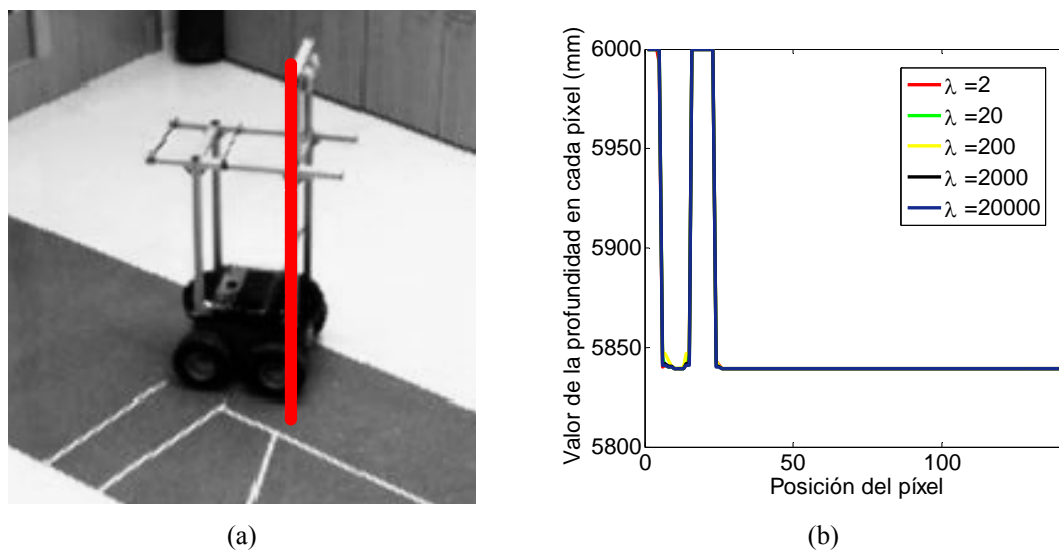


Figura 7.9. (a)Detalle de la imagen de entrada sobre la que se han resaltado en color rojo los píxeles para los que se representa el valor de la profundidad. (b) Valor de la profundidad en los píxeles marcados en rojo en (a) en función del valor de la constante  $\mu$ .

Por otro lado, en la Tabla 7.6 se presentan los valores máximos de la diferencia entre la profundidad calculada para  $\lambda=20000$ , y la obtenida con  $\lambda=[2, 20, 2000 \text{ y } 2000]$  para 5 imágenes pertenecientes a la secuencia de imágenes de prueba. Se puede observar que, tal como era de esperar, los valores mostrados en esta tabla son muy pequeños, y no superan en ningún caso los 80 milímetros. Esto es porque, como se ha comentado, la constante  $\lambda$  no aparece en la ecuación de descenso por el gradiente utilizada para la actualización de la profundidad, siendo su efecto debido únicamente a la actualización de los contornos de la segmentación en cada iteración.

Imagen	Valor máximo de la diferencia entre la profundidad calculada con $\lambda=20000$ y la obtenida con otros valores de $\lambda$ (milímetros)			
	$\lambda=2$	$\lambda=20$	$\lambda=200$	$\lambda=2000$
Imagen 1	13.5879	9.2532	10.3211	1.1841
Imagen 2	6.5273	6.4937	8.9864	2.6191
Imagen 3	77.5578	77.5578	77.5578	77.5578
Imagen 4	66.9806	66.9806	66.9806	59.2069
Imagen 5	74.4121	74.4121	61.9481	67.3497

Tabla 7.6. Valor máximo de la diferencia entre la profundidad, en mm, calculada con  $\lambda = 20000$ , y la obtenida con  $\lambda = [2, 20, 200, 2000]$ .

Dado que este valor es pequeño comparado con la distancia del robot a la cámara (ya que las cámaras se encuentran fijas en posiciones alejadas del robot, a una distancia

del orden de 4 metros) la variación de la profundidad  $Z_c$  no se considera determinante para la elección del valor de la constante  $\lambda$ .

**7.2.2.2. Efecto del valor de la constante  $\lambda$  sobre las curvas que definen las regiones en las que se segmenta la imagen**

Para determinar la influencia del valor de la constante  $\lambda$  sobre la evolución de las curvas, se ha evaluado la diferencia entre la curva obtenida tras la minimización de energía, para cada uno de los valores de  $\lambda$ , y la curva inicial calculada utilizando la técnica descrita en el capítulo 5. De esta forma, por cada una de las regiones se define un factor de evolución de las curvas ( $\delta\Phi_k$ ) calculado como el número de píxeles en que difieren las curvas iniciales y las obtenidas tras la minimización de energía, normalizado por el número de píxeles de la imagen ( $m$  filas) $\times$ ( $n$  columnas) según se puede ver en la ecuación (7.11).

$$\delta\Phi_k = \frac{n^\circ \text{ píxeles diferentes}}{m \cdot n} 100\% \quad k = 1, \dots, N - 1 \quad (7.11)$$

Si el factor de evolución de las curvas se divide entre el número de iteraciones se obtiene, para cada valor de la constante  $\lambda$ , una medida del cambio medio por iteración.

En la Tabla 7.7 se muestra: la diferencia (en número de píxeles) entre la curva inicial obtenida y la obtenida tras la minimización de energía, y el factor de evolución de las curvas, definido según la ecuación (7.11). Además, el número de píxeles se ha dividido entre el número de iteraciones (mostrado también en la segunda columna de la tabla) obteniéndose un valor de evolución media por cada iteración (en píxeles). Este valor se muestra en la última columna de la Tabla 7.7.

Valor de $\lambda$	Número de Iteraciones	Diferencia con la curva inicial (píxeles)	Factor de evolución $\delta\Phi_k$ (%)	Evolución media por cada iteración (píxeles)
$2 \cdot 10^{-1}$	27	1235	0.4020	45.7407
$2 \cdot 10^0$	30.36	1107	0.3604	36.4625
$2 \cdot 10^1$	34.64	780	0.2539	22.5173
$2 \cdot 10^2$	43.92	314	0.1022	7.1494
$2 \cdot 10^3$	55.60	62	0.0202	1.1151
$2 \cdot 10^4$	72.72	5	0.0016	0.0688

Tabla 7.7. Número de iteraciones, diferencia entre la curva inicial y el resultado de la segmentación, factor de evolución y evolución media de la curva por cada iteración, para  $\lambda = [2 \cdot 10^{-1}, 2 \cdot 10^0, 2 \cdot 10^1, 2 \cdot 10^2, 2 \cdot 10^3 \text{ y } 2 \cdot 10^4]$

En la Tabla 7.7 se observa que, a medida que se incrementa el valor de la constante  $\lambda$ , la diferencia entre la curva inicial y la obtenida tras la minimización de energía, disminuye. Sin embargo, el número de iteraciones para alcanzar la convergencia aumenta. Esto se debe a que cuanto mayor sea el valor de  $\lambda$ , mayor es la importancia del término asociado a la curvatura ( $\kappa_{\gamma_k}$ ), frente al término dependiente de la restricción de brillo 3D, en la ecuación de descenso por curvas de nivel.

En la Figura 7.10 se muestra un detalle de las curvas que definen el contorno de un robot móvil, obtenidas con diferentes valores de la constante  $\lambda$ , para una de las imágenes de la secuencia. En esta figura se observa que todas las curvas se aproximan al contorno del robot (ya que las curvas iniciales se encuentran ya próximas a dicho contorno), sin embargo, a medida que disminuye el valor de  $\lambda$ , las curvas se aproximan más al contorno real, debido a que las curvas presentan una evolución más acusada. Por tanto es deseable que la constante  $\lambda$  tome valores pequeños.

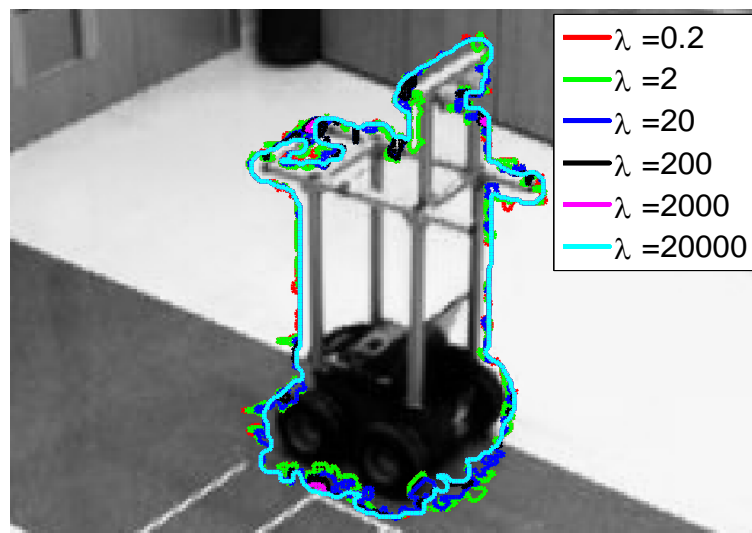


Figura 7.10. Detalle de las curvas obtenidas para una imagen de la secuencia utilizando diferentes valores de la constante  $\lambda$ .

### 7.2.2.3. Efecto del valor de la constante $\lambda$ sobre la velocidad 3D

Para finalizar, se analiza el efecto del valor de la constante  $\lambda$  sobre las componentes de velocidad 3D. Para ello se comparan los resultados obtenidos tras el proceso de minimización de la función objetivo, con las medidas de los sensores odométricos del robot. Al igual que en el análisis realizado para la elección de la constante  $\mu$ , las velocidades se comparan en el sistema de referencia global  $\Gamma_w$  debido a que las medidas del robot se encuentran representadas en ese sistema de referencia. La proyección de las

velocidades obtenidas tras la minimización de energía en  $\Gamma_c$ , en el sistema de referencia global  $\Gamma_w$  se realiza utilizando (7.8) y (7.9).

En la Tabla 7.8 se recogen los valores del error cuadrático medio de la velocidad lineal (en milímetros/segundo) y angular (en grados/segundo) calculado para un conjunto de 100 imágenes de prueba. En la Tabla 7.8 se aprecia que el error cuadrático medio tiene un mínimo para  $\lambda=2\cdot 10^2$ , y toma un valor similar para  $\lambda=2\cdot 10^3$ .

Valor de la constante $\lambda$	Número medio de iteraciones	Error cuadrático medio de la velocidad lineal (mm/s)	Error cuadrático medio de la velocidad angular (grados/s)
$2\cdot 10^{-1}$	28.04	37.406	1.6221
$2\cdot 10^0$	27.68	36.793	1.6322
$2\cdot 10^1$	23.04	32.4753	1.6477
$2\cdot 10^2$	21.16	30.2123	1.6459
$2\cdot 10^3$	18.56	30.3548	1.6952
$2\cdot 10^4$	19	31.8055	1.6930

Tabla 7.8. Error cuadrático medio, en función del valor de la constante  $\lambda$ , cometido en la estimación de la velocidad lineal y angular, y número medio de iteraciones necesarias hasta la convergencia del algoritmo para las 100 imágenes de la secuencia de prueba (a la que pertenecen las imágenes de la Figura 7.1) para valores de  $\lambda$  entre  $2\cdot 10^{-1}$  y  $2\cdot 10^4$ .

A la vista de los resultados expuestos, se ha determinado que el orden de magnitud de la constante  $\lambda$  debe ser de  $10^2$ , ya que entonces el error cuadrático medio de la velocidad es mínimo. Con objeto de obtener un rango de valores adecuados para la constante  $\lambda$ , se han evaluado el número de iteraciones, el error en la velocidad y las curvas iniciales para las 100 imágenes de prueba, tras realizar la minimización de energía con un conjunto de valores de  $\lambda$  entre 100 y 2000. En la Tabla 7.9 se muestra el número de iteraciones y el error cuadrático medio de las componentes de velocidad lineal y angular obtenido para 100 imágenes, para cada uno de los valores de la constante  $\lambda$  considerados. En esta tabla se puede observar que el valor para el que tanto el error cuadrático medio de la velocidad lineal, como el número de iteraciones son menores es  $\lambda= 5\cdot 10^2$ . En la Tabla 7.9 también se puede observar que para cualquiera de los valores de  $\lambda$  considerados (entre 100 y 2000) los resultados obtenidos son similares.

Valor de la constante $\lambda$	Número medio de iteraciones	Error cuadrático medio de la velocidad lineal (mm/s)	Error cuadrático medio de la velocidad angular (grados/s)
$\lambda = 1 \cdot 10^2$	20.84	29.6361	1.6383
$\lambda = 2 \cdot 10^2$	21.16	30.2123	1.6459
$\lambda = 5 \cdot 10^2$	18.28	27.4717	1.6910
$\lambda = 1 \cdot 10^3$	19.04	29.0265	1.6974
$\lambda = 1.5 \cdot 10^3$	19.28	30.1799	1.6934
$\lambda = 2 \cdot 10^3$	18.56	30.3548	1.6952

Tabla 7.9. Valor medio del error cometido en la estimación de la velocidad lineal y número medio de iteraciones necesarias hasta la convergencia del algoritmo para las 100 imágenes de la secuencia de prueba en función del valor de la constante  $\lambda$ , para valores de  $\lambda$  entre 100 y 2000.

Los resultados presentados en la Tabla 7.9 demuestran que cualquier valor de la constante  $\lambda$  perteneciente al rango propuesto (entre 100 y 2000) es válido para la segmentación de movimiento, ya que el algoritmo de minimización presenta una baja sensibilidad respecto al valor de  $\lambda$  en ese rango. Esto se debe a que las curvas iniciales obtenidas mediante el proceso de inicialización explicado en el capítulo 5 se encuentran próximas a los contornos reales de los robots móviles presentes en las imágenes.

### 7.2.3. Validación de los valores de $\lambda$ y $\mu$ elegidos

A lo largo de los apartados 7.2.1 y 7.2.2 se ha realizado un análisis del efecto de cada una de las constantes  $\lambda$  y  $\mu$  en el proceso de minimización de energía para las imágenes de la secuencia 1 (a la que pertenecen las imágenes de la Figura 7.1). En todos los casos mostrados en los apartados anteriores se ha estudiado cada una de las constantes por separado, obteniendo como resultado un rango de posibles valores para cada una de ellas. En este apartado se realiza un análisis del efecto que tienen ambas constantes sobre el proceso de minimización de energía evaluando, para ello, tanto el error cuadrático medio de las componentes de velocidad lineal y angular, como el número de iteraciones necesarias hasta la convergencia del algoritmo.

Además de las imágenes pertenecientes a la secuencia 1 (Figura 7.1), los valores de  $\lambda$  y  $\mu$  elegidos se han utilizado para la segmentación de movimiento y obtención de la posición 3D en otras dos secuencias de imágenes, captadas también en el espacio inteligente, en diferentes condiciones de iluminación. En la Figura 7.2(a) se muestra una imagen perteneciente a la secuencia 2, mientras que la imagen de la Figura 7.2(b) corresponde a la secuencia 3.



En la Tabla 7.10 se recogen los resultados obtenidos. No se incluye el error cuadrático medio de la velocidad angular debido a que en los apartados anteriores se ha comprobado que la variación de este error es poco significativa. En la Tabla 7.10 se observa que el error cuadrático medio de la velocidad lineal es inferior a 35 mm/s para las tres secuencias consideradas. Cabe destacar que el peor caso se da para la secuencia 1 (que ha sido la utilizada para la elección de los valores de las constantes) siendo mejores para las secuencias 2 y 3. Además, en esta tabla se observa que, en general, el valor de la constante  $\lambda$  apenas influye en el valor del error. Por otro lado, el número de iteraciones varía entre 2 y 32.

		Numero medio de iteraciones			Error cuadrático medio de la velocidad lineal (mm/s)		
		$\lambda=100$	$\lambda=500$	$\lambda=1000$	$\lambda=100$	$\lambda=500$	$\lambda=1000$
Secuencia 1	$\mu = 1 \cdot 10^{-1}$	26.24	28.84	31.88	22.5622	22.6922	25.2050
	$\mu = 3 \cdot 10^{-1}$	22.04	21.28	21.76	27.6377	25.6458	27.0687
	$\mu = 5 \cdot 10^{-1}$	20.84	18.28	19.04	29.6361	27.4717	29.0265
	$\mu = 7 \cdot 10^{-1}$	19.96	17.12	17.32	30.6102	28.3763	29.9305
	$\mu = 1$	19.84	16.52	15.96	32.4779	30.1242	31.1992
Secuencia 2	$\mu = 1 \cdot 10^{-1}$	2.00	2.00	2.00	24.2824	24.2824	24.2824
	$\mu = 3 \cdot 10^{-1}$	6.72	6.72	6.72	24.2903	24.2903	24.2903
	$\mu = 5 \cdot 10^{-1}$	10.64	10.64	10.64	24.2792	24.2792	24.2792
	$\mu = 7 \cdot 10^{-1}$	12.56	12.56	12.56	24.2779	24.2779	24.2779
	$\mu = 1$	13.96	13.96	13.96	24.2783	24.2783	24.2783
Secuencia 3	$\mu = 1 \cdot 10^{-1}$	20.48	14.44	10.56	19.8448	21.5414	21.5718
	$\mu = 3 \cdot 10^{-1}$	19.60	15.12	11.80	19.9615	20.3826	19.5541
	$\mu = 5 \cdot 10^{-1}$	19.68	14.72	12.08	19.9753	19.2207	19.0097
	$\mu = 7 \cdot 10^{-1}$	19.96	14.52	12.24	20.7359	19.0234	18.5847
	$\mu = 1$	20.08	13.92	11.72	21.4294	19.1651	18.4020

Tabla 7.10. Número medio de iteraciones, y error cuadrático medio de la velocidad lineal (en mm/s) en función del valor de las constantes  $\lambda$  y  $\mu$ . Los resultados se obtienen para 25 imágenes de las secuencias 1 (Figura 7.1), 2 (Figura 7.2(a)) y 3 (Figura 7.2 (b)).

Los resultados mostrados en la Tabla 7.10 demuestran que los valores de  $\lambda$  y  $\mu$  elegidos en los apartados 7.2.1 y 7.2.2 para las imágenes pertenecientes a la secuencia 1 (Figura 7.1) son igualmente válidos para las imágenes de las secuencias 2 y 3 (mostradas en la Figura 7.2(a) y (b) respectivamente), a pesar de que estas secuencias han sido adquiridas con diferentes condiciones de iluminación.

Cabe resaltar que los resultados obtenidos para las secuencias 2 y 3, son incluso mejores que los logrados para la secuencia 1 (que ha sido la utilizada en el proceso de elección de las constantes  $\lambda$  y  $\mu$ ).

### 7.3. Valores de las constantes $\lambda$ y $\mu$ para la función objetivo definida para múltiples cámaras

Al igual que en el caso de una cámara, la función objetivo, (7.2), obtenida para la segmentación de movimiento y la obtención de la posición 3D, a partir de las imágenes captadas por  $n_c$  cámaras, está formada por tres términos. La contribución de cada uno de estos términos viene determinada por los valores de las constantes reales y positivas,  $\lambda$  y  $\mu$ . En concreto, la constante  $\mu$  pondera la contribución del término asociado al gradiente de la profundidad ( $Z_{ci}$ ) en el sistema de referencia de cada una de las  $n_c$  cámaras, mientras que la constante  $\lambda$  establece el peso del término ligado a las curvas  $\{\gamma_{ki}\}_{k=1, \dots, N-1}^{i=1, \dots, n_c}$  que definen los contornos de las  $N$  regiones en las que se ha dividido cada una de las imágenes.

A continuación se analiza el efecto de cada una de las constantes sobre los resultados de la minimización de energía. Para todas las pruebas experimentales presentadas a continuación se ha utilizado la función de preservación de bordes cuadrática ( $g(a) = a^2$ ) ya que es la función más sencilla y la que consume un menor tiempo de cómputo. Además, según se ha demostrado en el apartado 7.1, los resultados obtenidos con las diferentes funciones de preservación de bordes son similares si se ajusta el valor de la constante  $\mu$  de forma adecuada para cada caso.

En los siguientes apartados se presentan las pruebas experimentales realizadas, así como los resultados obtenidos en el proceso de ajuste de las constantes de ponderación  $\lambda$  y  $\mu$ . En primer lugar se realiza un ajuste de la constante  $\mu$  para la función cuadrática de preservación de bordes, a continuación se realiza el ajuste de la constante  $\lambda$ , y se finaliza con la validación conjunta de los valores elegidos.

#### 7.3.1. Ajuste de la constante $\mu$

Como ya se ha comentado, la constante  $\mu$  pondera la contribución del término asociado al gradiente de la profundidad en cada  $\Gamma_{ci}$  ( $i=1, \dots, n_c$ ) a la función objetivo, (7.2). En el proceso de minimización de energía, esta constante aparece en la ecuación

de descenso por el gradiente de la profundidad. En esta ecuación, la constante  $\mu$  determina el peso del segundo término (dependiente del gradiente de la profundidad), frente a la restricción de brillo. Además, de igual forma que en el caso de una cámara, el valor de la constante  $\mu$  está directamente relacionado con la función de preservación de bordes ( $g(a)$ ) elegida. Como ya se ha comentado, para las pruebas realizadas en este apartado se ha elegido la función cuadrática ( $g(a) = a^2$ ) para la preservación de bordes.

Para la obtención del valor de la constante  $\mu$  se ha realizado un conjunto de pruebas experimentales utilizando 100 imágenes pertenecientes a la secuencia 1 (Figura 7.1(a)). En estas pruebas se han obtenido los resultados de la segmentación de movimiento de las 100 imágenes considerando diferentes valores de la constante  $\mu$  entre  $10^{-3}$  y 5.

En cada una de estas pruebas se ha evaluado la calidad de la segmentación resultante, así como el error cometido en el cálculo de las velocidades. Este error se ha calculado, en todos los casos, como la diferencia entre las velocidades estimadas por los sensores odométricos del robot y las obtenidas tras la minimización de la función objetivo. Además, también se ha tenido en cuenta el número de iteraciones necesarias hasta la convergencia del algoritmo.

Para la realización de todas estas pruebas se ha considerado un valor de 20.000 para la constante  $\lambda$ , ya que al inicializar las curvas en posiciones cercanas a los contornos de los robots móviles, utilizando la técnica expuesta en el capítulo 5, no es necesario que se produzcan grandes cambios en las curvas durante el proceso de minimización de energía. Además, con objeto de que la evolución de las curvas sea lenta, se fija  $\lambda$  a un valor elevado (20.000 en este caso). De esta forma se trata de independizar el efecto de la constante  $\mu$  del de la constante  $\lambda$ .

A continuación se evalúa el efecto de la constante  $\mu$  sobre cada uno de los grupos de variables implicados en la función objetivo definida en la ecuación (7.2): las curvas que definen los contornos de los robots móviles, la profundidad, y las componentes de velocidad 3D.

No se han incluido resultados para valores de  $\mu$  superiores a 1, debido a que se ha comprobado que para dichos valores, al utilizar la función cuadrática de preservación de bordes, el proceso de minimización de energía no alcanza la convergencia.

### 7.3.1.1. Efecto del valor de la constante $\mu$ sobre las curvas que definen las regiones en que se segmenta la imagen

En las pruebas experimentales, realizadas con los diferentes valores de la constante  $\mu$  considerados, se ha puesto de manifiesto que la influencia de dicha constante sobre las curvas que definen los contornos de las diferentes regiones en que se divide la imagen es poco significativa. En la Figura 7.11 se muestra la evolución del número de píxeles contenidos por la curva que define el contorno del robot móvil para 5 de las 100 imágenes de la secuencia considerada. En esta figura se muestran los resultados obtenidos para 4 cámaras calibradas. Se puede observar como en todos los casos la variación del número de píxeles en función del valor de la constante  $\mu$  es muy pequeña.

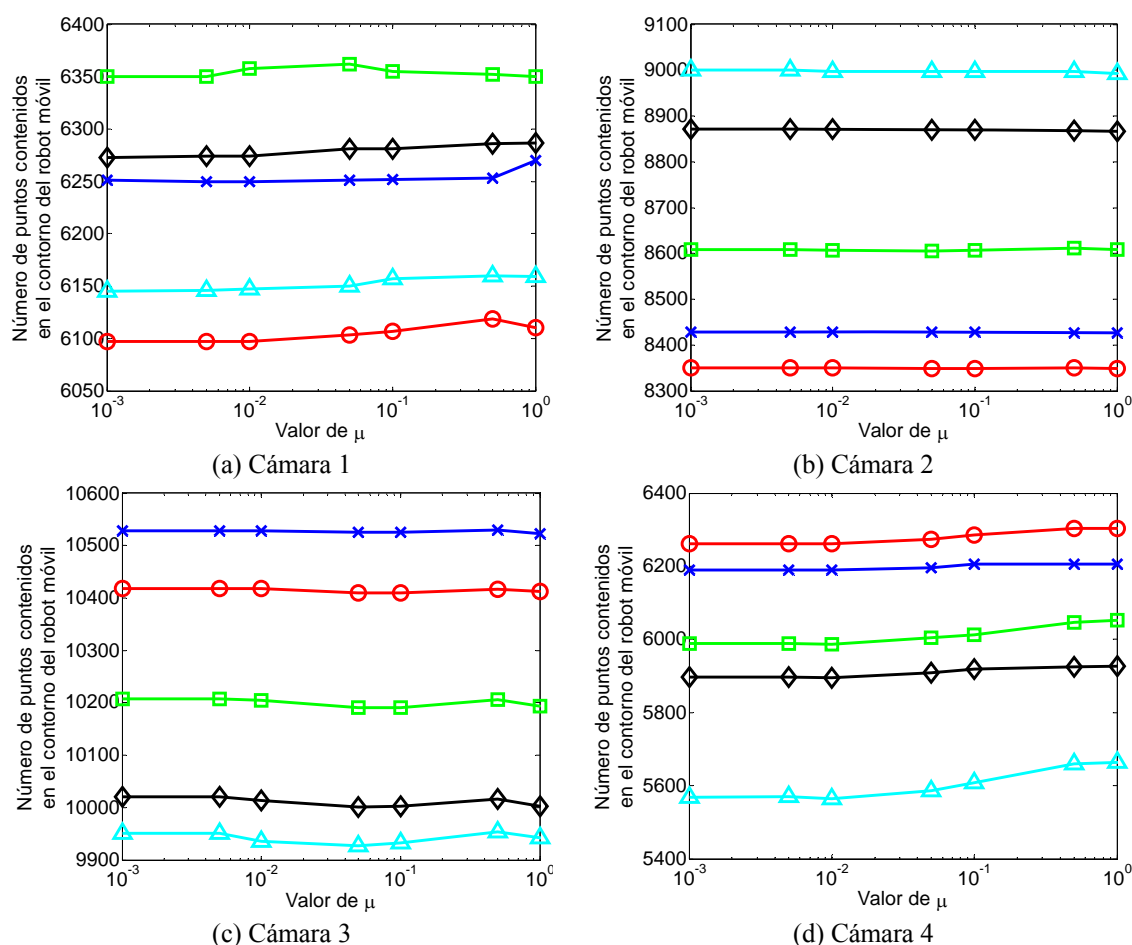


Figura 7.11. Número de píxeles contenidos por la curva que define el contorno del robot móvil en función del valor de la constante  $\mu$ , para 5 de las 100 imágenes de la secuencia de prueba. (a) Número de píxeles por la curva obtenida para la imagen de la cámara 1. (b) Número de píxeles en la curva obtenida para la imagen de la cámara 2. (c) Número de píxeles en la curva obtenida para la imagen de la cámara 3. (d) Número de píxeles en la curva obtenida para la imagen de la cámara 4.

Este mismo afecto se aprecia en la Figura 7.12, en la que se ha representado un detalle de las curvas que definen el contorno del robot móvil para una de las imágenes de la secuencia. En la Figura 7.12 únicamente se han representado las curvas para las

imágenes captadas por las cámaras 1 y 3 debido a que para las otras dos cámaras disponibles los resultados obtenidos son similares.

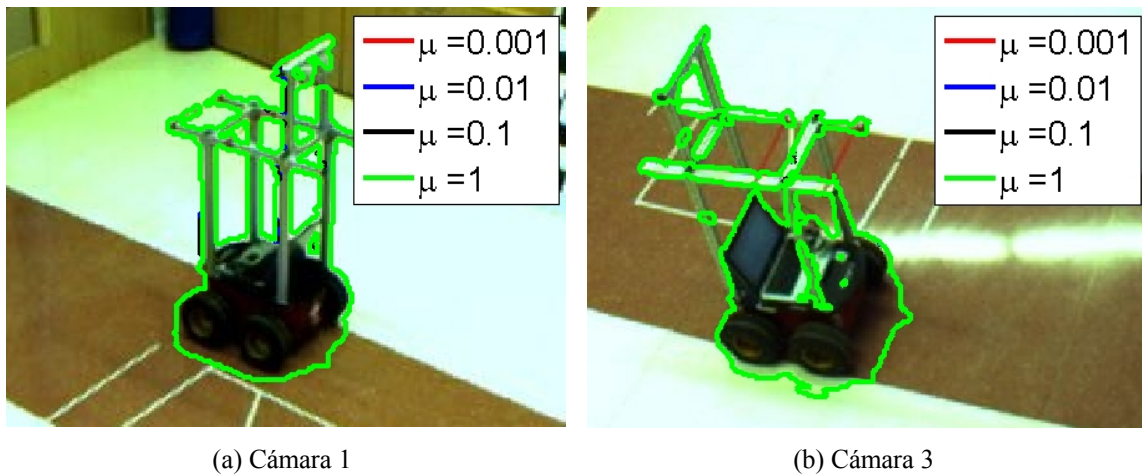


Figura 7.12. Detalle de las curvas iniciales de una de las imágenes de la secuencia (a) Vista desde la cámara 1 (b) Vista desde la cámara 3.

Los resultados mostrados en estas figuras demuestran que el valor elegido para la constante  $\mu$  no afecta a las curvas iniciales obtenidas, ya que todas las curvas aparecen superpuestas. Esto se debe al valor de  $\lambda$  elegido, ya que, aunque a medida que se incrementa el valor de  $\mu$  se ejecuta un número mayor de iteraciones hasta la convergencia, al elegir para  $\lambda$  un valor elevado, las curvas apenas se modifican en cada iteración.

### 7.3.1.2. Efecto del valor de la constante $\mu$ sobre la profundidad

Para evaluar el efecto del valor de la constante  $\mu$  sobre la profundidad se ha calculado el valor medio de la profundidad de los puntos interiores a la curva que define el contorno del robot móvil. En la Tabla 7.11 se presentan los resultados obtenidos para una de las imágenes de la secuencia 1. Para algunos de los valores de  $\mu$  no se muestran resultados. Esto se debe a que para esos valores, la profundidad media estimada se encuentra fuera del rango de valores permitidos, es decir es negativo o bien supera el valor máximo permitido (impuesto por las dimensiones del ISPACE-UAH).

En la Tabla 7.11 se observa que para valores de la constante  $\mu$  inferiores a  $5 \cdot 10^{-2}$ , el valor medio de la profundidad disminuye a medida que aumenta el valor de  $\mu$ . Sin embargo, dado que esa variación es poco significativa, cualquiera de los valores de  $\mu$  entre  $10^{-3}$  y  $5 \cdot 10^{-2}$  es válido para la segmentación de movimiento y la obtención de la posición 3D. Por otro lado, para valores de  $\mu$  superiores a  $5 \cdot 10^{-2}$ , a pesar de que se

cumple el criterio de convergencia, los valores de profundidad obtenidos no son válidos. Esto impone una restricción al valor de  $\mu$ , que debe ser siempre inferior a  $10^{-1}$  para garantizar la validez de los resultados obtenidos.

Valor de $\mu$	Valor medio de la profundidad en los puntos interiores a la curva (mm)			
	Cámara 1	Cámara 2	Cámara 3	Cámara 4
$\mu = 10^{-3}$	5717.6306	4347.1036	4348.0277	6152.7348
$\mu = 5 \cdot 10^{-3}$	5712.8359	4341.8772	4343.1084	6149.1143
$\mu = 10^{-2}$	5706.9014	4335.4288	4337.0280	6144.6068
$\mu = 5 \cdot 10^{-2}$	5544.2520	4169.2482	4176.0461	6009.8678
$\mu = 10^{-1}$	-	-	-	-
$\mu = 5 \cdot 10^{-1}$	-	-	-	-
$\mu = 1$	-	-	-	-

Tabla 7.11. Valor medio de la profundidad de los puntos contenidos en la curva que define el contorno del robot móvil para una imagen de la secuencia 1 (a la que pertenecen las imágenes de la Figura 7.1) en función del valor de la constante  $\mu$ .

Cabe destacar que, aunque en la Tabla 7.11 únicamente se muestran los resultados obtenidos para una de las 100 imágenes de la secuencia, en las pruebas experimentales se ha comprobado que en todos los casos los resultados obtenidos son similares: pequeñas variaciones de la profundidad para los diferentes valores de la constante  $\mu$  hasta un determinado valor de  $\mu$  a partir del cual los resultados están fuera del rango permitido. El valor de  $\mu$  para el que se produce este fenómeno varía ligeramente para cada imagen, sin embargo, para todas ellas se ha comprobado que si el valor de  $\mu$  es inferior a  $5 \cdot 10^{-2}$  se obtienen resultados válidos para la profundidad. Es decir, para la secuencia de 100 imágenes de prueba, el valor de  $\mu$  debe ser siempre inferior a  $5 \cdot 10^{-2}$ .

### 7.3.1.3. Efecto del valor de la constante $\mu$ sobre la velocidad 3D y el número de iteraciones hasta la convergencia

Finalmente, se analiza el efecto del valor elegido para la constante  $\mu$  sobre las componentes de velocidad 3D. Para ello se comparan los resultados obtenidos tras el proceso de minimización de la función objetivo con las medidas de los sensores odométricos del robot. Esta comparación se realiza en el sistema de referencia global  $\Gamma_w$  respecto al que están expresadas tanto las velocidades estimadas por el algoritmo, como las medidas por los sensores odométricos del robot.

Para evaluar el efecto de la constante  $\mu$  sobre las velocidades, se obtiene el error cuadrático medio para 100 imágenes de la secuencia de prueba (a la que pertenecen las imágenes de la Figura 7.1). En la Tabla 7.12 se presentan los resultados obtenidos. En la segunda columna de esta tabla se muestra, para 100 imágenes de la secuencia, el número medio de iteraciones hasta la convergencia, mientras que en la tercera y cuarta columna se muestra el error cuadrático medio de la velocidad lineal y angular, respectivamente, calculados para 100 imágenes de la secuencia 1. En esta tabla se observa que, a medida que se incrementa el valor de  $\mu$ , aumenta también el número de iteraciones hasta la convergencia (llegando a no converger para  $\mu=1$ ). Respecto al error cuadrático medio de la velocidad lineal, el valor más pequeño de este error se obtiene para  $\mu=5 \cdot 10^{-1}$ , sin embargo, este valor no se considera adecuado debido a que requiere un alto número de iteraciones. Esto implica que es necesario alcanzar una solución de compromiso entre el número de iteraciones y el error de las velocidades. A la vista de los resultados mostrados en la Tabla 7.12, el valor en el que se tiene una mejor relación entre el error en las velocidades y el número de iteraciones es  $\mu=10^{-2}$ .

Valor de la constante $\mu$	Número medio de iteraciones	Error cuadrático medio de la velocidad lineal (mm/s)	Error cuadrático medio de la velocidad angular (grados/s)
$\mu = 10^{-3}$	3.48	16.3493	1.4440
$\mu = 5 \cdot 10^{-3}$	3.52	16.0946	1.4450
$\mu = 10^{-2}$	5.76	12.6646	1.5525
$\mu = 5 \cdot 10^{-2}$	17.44	14.3699	1.6840
$\mu = 10^{-1}$	77.48	23.6124	1.6774
$\mu = 5 \cdot 10^{-1}$	160.28	7.4013	1.6774
$\mu = 1$	No converge	-	-

Tabla 7.12. Valor medio del error cometido en la estimación de la velocidad lineal y número medio de iteraciones necesarias hasta la convergencia del algoritmo para las 100 imágenes de la secuencia de prueba (a la que pertenece la imagen de la Figura 7.1) en función del valor de la constante  $\mu$ , para valores de  $\mu$  entre  $10^{-3}$  y 1.

Partiendo de los resultados mostrados en la Tabla 7.4, se ha realizado un nuevo análisis para valores de  $\mu$  entre  $5 \cdot 10^{-3}$  y  $10^{-1}$ , con objeto de determinar en qué rango de valores de  $\mu$  se obtiene una mejor relación entre el número de iteraciones y el error cuadrático medio en la velocidad lineal para las 100 imágenes de prueba. Los resultados obtenidos se presentan en la Tabla 7.13.

Valor de la constante $\mu$	Número medio de iteraciones	Error cuadrático medio de la velocidad lineal (mm/s)	Error cuadrático medio de la velocidad angular (grados/s)
$\mu = 3 \cdot 10^{-3}$	16.2669	3.5200	1.4419
$\mu = 5 \cdot 10^{-3}$	16.0946	3.5200	1.4450
$\mu = 7 \cdot 10^{-3}$	14.7660	4.3200	1.4893
$\mu = 10^{-2}$	12.6646	5.7600	1.5525
$\mu = 3 \cdot 10^{-2}$	11.3235	5.2400	1.6908
$\mu = 5 \cdot 10^{-2}$	14.3699	17.4400	1.6840
$\mu = 7 \cdot 10^{-2}$	21.2942	27.8000	1.6779
$\mu = 10^{-1}$	77.4800	23.6124	1.6774

Tabla 7.13. Valor medio del error cometido en la estimación de la velocidad lineal y número medio de iteraciones necesarias hasta la convergencia del algoritmo para las 100 imágenes de la secuencia de prueba (a la que pertenecen las imágenes de la Figura 7.1) en función del valor de la constante  $\mu$ , para valores de  $\mu$  entre  $3 \cdot 10^{-3}$  y  $10^{-1}$ .

Como ya se ha comentado, es necesario alcanzar una solución de compromiso entre el número de iteraciones y el error cometido en el cálculo de la velocidad. En la Tabla 7.13 se observa que, para las imágenes de prueba consideradas, el error cuadrático medio de la velocidad lineal presenta su valor mínimo para  $\mu=3 \cdot 10^{-3}$  (aumentando a partir de dicho valor), mientras que el mínimo número de iteraciones hasta la convergencia se produce para  $\mu=3 \cdot 10^{-2}$ . A la vista de los resultados, el rango de valores de la constante  $\mu$  para los que se obtienen una mejor relación entre el número medio de iteraciones y el error cuadrático medio de la velocidad lineal y angular es  $\mu \in [3 \cdot 10^{-3}, 3 \cdot 10^{-2}]$ . Se elegirá un valor más elevado de  $\mu$  cuando mayor prioridad se le dé al número de iteraciones frente al error en la velocidad. Cabe destacar que estos valores cumplen la restricción impuesta por la profundidad, ya que todos los valores incluidos en el rango determinado para la constante  $\mu$  son inferiores a  $5 \cdot 10^{-2}$ .

### 7.3.2. Ajuste de la constante $\lambda$

Para el ajuste del valor de la constante  $\lambda$  para las imágenes de prueba ha sido realizado un conjunto de pruebas experimentales. En ellas se han evaluado los resultados de la segmentación de movimiento y el posicionamiento 3D de un robot móvil a partir de 100 imágenes captadas en el espacio inteligente. Al igual que en el caso del ajuste de la constante  $\mu$ , en cada una de las pruebas experimentales realizadas se ha evaluado, tanto la calidad de la segmentación resultante, como el error cometido en el cálculo de las velocidades (calculado como la diferencia entre las velocidades



estimadas por los sensores odométricos del robot, y las obtenidas tras la minimización de la función objetivo). Además, también se ha tenido en cuenta el número de iteraciones necesarias hasta la convergencia del algoritmo. En todas las pruebas realizadas para el ajuste de la constante  $\lambda$ , el valor de  $\mu$  se ha fijado a  $3 \cdot 10^{-2}$ , debido a que en el apartado 7.3.1 se ha comprobado que dicho valor de  $\mu$  proporciona resultados adecuados con un número pequeño de iteraciones.

A continuación se evalúa el efecto de la constante  $\lambda$  sobre cada uno de los grupos de variables implicados en la función objetivo definida en (7.1): las curvas que definen los contornos de los robots móviles, la profundidad, y las componentes de velocidad 3D.

### 7.3.2.1. Efecto del valor de la constante $\lambda$ sobre la profundidad

Para evaluar el efecto del valor de la constante  $\lambda$  sobre la profundidad ( $Z_c$ ) se ha calculado el valor medio de esta variable en los píxeles contenidos en la curva que define el contorno del robot móvil para los diferentes valores de  $\lambda$ . Los resultados se muestran en la Tabla 7.14. En esta tabla se observa que, aunque el valor de la profundidad media aumenta con el incremento del valor de  $\lambda$ , esto no permite determinar qué valores de la constante proporcionan mejores resultados. Esto es debido a que la variación de la medida de profundidad no se debe tanto al valor de  $\lambda$  como al número de iteraciones que se ejecutan hasta la convergencia para cada uno de los valores de  $\lambda$ . Por este motivo, la evolución de la profundidad media para los diferentes valores de  $\lambda$  no es un parámetro decisivo para la elección del valor de esta constante de ponderación.

Valor de $\lambda$	Valor medio de la profundidad en los puntos interiores a la curva (mm)			
	Cámara 1	Cámara 2	Cámara 3	Cámara 4
$2 \cdot 10^{-1}$	4809.3789	3804.1211	3729.5504	5573.3793
$2 \cdot 10^0$	5029.6857	3875.3591	3811.5224	5685.7867
$2 \cdot 10^1$	5095.7677	3944.7846	3874.9303	5740.0241
$2 \cdot 10^2$	5305.9126	4030.9369	4005.4272	5867.1077
$2 \cdot 10^3$	5626.6434	4246.4093	4246.3704	6075.3297
$2 \cdot 10^4$	5650.8136	4276.3271	4280.8515	6100.7397

Tabla 7.14. Valor medio de la profundidad, en función del valor de la constante  $\lambda$ , de los puntos contenidos en la curva que define el contorno del robot móvil para una imagen de la secuencia 1.

### 7.3.2.2. Efecto del valor de la constante $\lambda$ sobre las curvas que definen las regiones en que se segmenta la imagen

Para determinar la influencia del valor de la constante  $\lambda$  sobre la evolución de las curvas se ha evaluado la diferencia entre la curva obtenida tras la minimización de energía para cada uno de los valores de  $\lambda$ , y la curva inicial calculada utilizando la técnica descrita en el capítulo 5. Para ello se utiliza el factor de evolución de las curvas ( $\delta\Phi_k$ ) definido en la ecuación (7.11) del apartado 7.2.2.

En la Tabla 7.15 se muestran los resultados obtenidos para cada una de las 4 cámaras disponibles. En esta tabla se recoge tanto el número medio de iteraciones hasta la convergencia, como la diferencia (en número de píxeles) entre la curva inicial y la curva obtenida tras el proceso de minimización de energía, y el factor de evolución ( $\delta\Phi_k$ ) definido en la ecuación (7.11). En la Tabla 7.15 se observa como a medida que aumenta el valor de la constante  $\lambda$ , tanto el número de iteraciones hasta la convergencia, como el número de píxeles diferentes entre la curva inicial y la final disminuyen.

El descenso del número de píxeles en que se modifican las curvas en el proceso de minimización de energía se debe tanto al incremento del valor de  $\lambda$  como al descenso del número de iteraciones.

Valor de $\lambda$		$2 \cdot 10^{-1}$	$2 \cdot 10^0$	$2 \cdot 10^1$	$2 \cdot 10^2$	$2 \cdot 10^3$	$2 \cdot 10^4$
<b>Número de medio iteraciones</b>		66.08	61.84	49.56	19.48	10.04	5.24
<b>Diferencia con la curva inicial (píxeles)</b>	Cam1	2063.12	1899.72	1420.72	922.40	470.32	23.80
	Cam2	4098.76	3888.32	3125.72	1880.12	739.72	16.32
	Cam3	4243.12	3856.44	2752.00	1379.40	389.16	8.12
	Cam4	2647.84	2459.80	1841.68	1125.92	520.76	66.68
<b>Factor de evolución <math>\delta\Phi_k</math> (%)</b>	Cam1	0.6716	0.6184	0.4625	0.3003	0.1531	0.0094
	Cam2	1.3342	1.2657	1.0175	0.6120	0.2408	0.0053
	Cam3	1.3812	1.2554	0.8958	0.4490	0.1267	0.0026
	Cam4	0.8619	0.8007	0.5995	0.3665	0.1695	0.0217

Tabla 7.15. Número de iteraciones, diferencia entre la curva inicial y el resultado de la segmentación y factor de evolución de la curva para  $\lambda = [2 \cdot 10^{-1}, 2 \cdot 10^0, 2 \cdot 10^1, 2 \cdot 10^2, 2 \cdot 10^3$  y  $2 \cdot 10^4]$ .

Finalmente, en la Figura 7.13 se muestra, de forma gráfica, la evolución del número de píxeles en que se diferencian la curva inicial y la final, en función del valor de  $\lambda$  para cada una de las 4 cámaras disponibles.

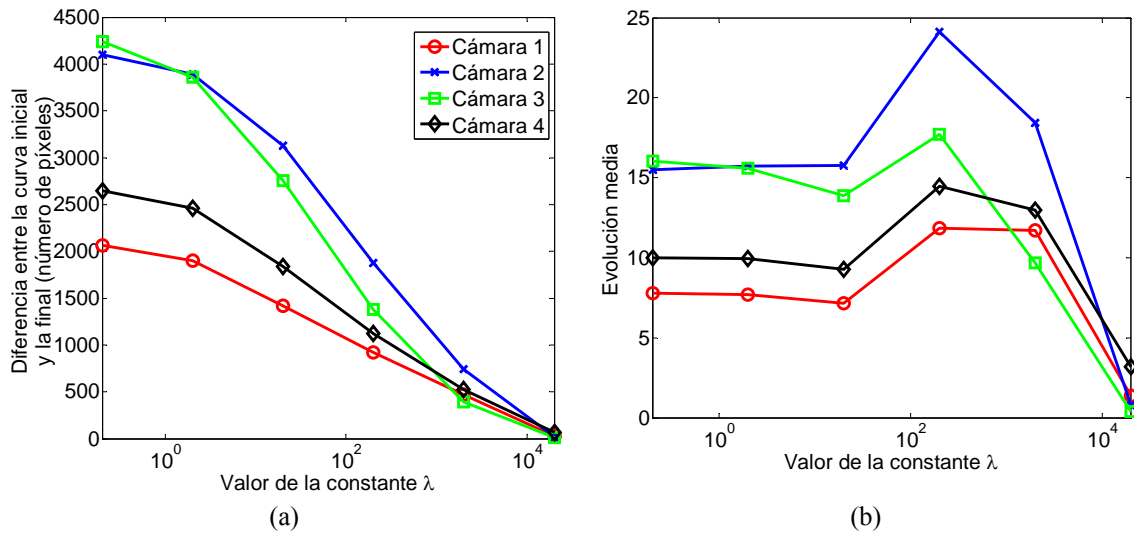


Figura 7.13. (a) Número de píxeles en que se diferencia la curva inicial de la curva obtenida tras la minimización de energía para cada una de las cuatro cámaras disponibles. (b) Evolución media de las curvas por cada iteración (calculada dividiendo el número de píxeles en que difieren la curva inicial y la final entre el número medio de iteraciones).

Tanto en la Tabla 7.15 como en la Figura 7.13 se aprecia que, a medida que aumenta el valor de la constante  $\lambda$ , la evolución de las curvas tiende a ser menor. Esto se debe tanto al descenso del número medio de iteraciones hasta la convergencia, como al incremento del valor de la constante de ponderación  $\lambda$ , ya que, cuanto mayor sea este valor, mayor es la influencia del término asociado a la curvatura en la iteración anterior de la ecuación de descenso por curvas de nivel (ecuación (4.80)).

En la Figura 7.14 se muestra un detalle de las curvas que definen el contorno de un robot móvil, obtenidas con diferentes valores de la constante  $\lambda$ , para una de las imágenes de la secuencia.

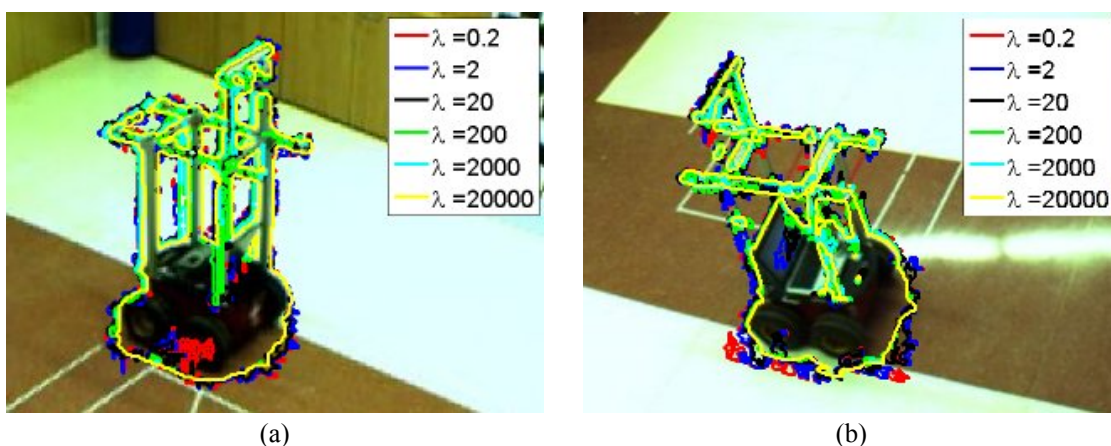


Figura 7.14. Detalle de las curvas obtenidas para una imagen de la secuencia utilizando diferentes valores de la constante  $\lambda$ . (a) Detalle de las curvas en la imagen captada por la cámara 1. (b) Detalle de las curvas en la imagen captada por la cámara 3.

En la Figura 7.14 se observa que todas las curvas se aproximan al contorno del robot, sin embargo, a medida que aumenta el valor de  $\lambda$  (con  $\lambda > 1$ ), las curvas

permanecen más próximas al contorno real del robot. Esto implica que, para el correcto ajuste de las curvas, el valor de  $\lambda$  debe ser elevado (entre 2000 y 20000).

### 7.3.2.3. Efecto del valor de la constante $\lambda$ sobre la velocidad 3D y el número de iteraciones hasta la convergencia

Para finalizar, se analiza el efecto del valor de la constante  $\lambda$  sobre las componentes de velocidad 3D. Para ello se comparan los resultados obtenidos tras el proceso de minimización de la función objetivo, con las medidas de los sensores odométricos del robot, todos ellos representados en el sistema de referencia del mundo  $\Gamma_w$  común para todas las cámaras.

En la Tabla 7.16 se recogen los valores del error cuadrático medio de la velocidad lineal (en milímetros/segundo) y angular (en grados/segundo) calculado para un conjunto de 100 imágenes de prueba pertenecientes a la secuencia de las imágenes mostradas en la Figura 7.1.

Valor de la constante $\lambda$	Número medio de iteraciones	Error cuadrático medio de la velocidad lineal (mm/s)	Error cuadrático medio de la velocidad angular (grados/s)
$2 \cdot 10^{-1}$	66.08	22.4875	1.6774
$2 \cdot 10^0$	61.84	22.2134	1.6776
$2 \cdot 10^1$	49.56	20.641	1.6789
$2 \cdot 10^2$	19.48	11.9737	1.6863
$2 \cdot 10^3$	10.04	10.4058	1.6823
$2 \cdot 10^4$	5.24	11.3235	1.6908

Tabla 7.16. Error cuadrático medio cometido en la estimación de la velocidad lineal y angular, y número medio de iteraciones necesarias hasta la convergencia del algoritmo para las 100 imágenes de la secuencia de prueba, en función del valor de la constante  $\lambda$ , para valores de  $\lambda$  entre  $2 \cdot 10^{-1}$  y  $2 \cdot 10^4$ .

En esta tabla se observa como, a medida que se incrementa el valor de  $\lambda$ , tanto el número de iteraciones, como el error cuadrático medio de la velocidad lineal disminuyen, mientras que el error cuadrático medio de la velocidad angular aumenta. Sin embargo, dado que la variación del error en la velocidad angular es insignificante (del orden de centésimas de grado/s) se ha determinado que el rango de valores de la constante  $\lambda$  debe encontrarse entre  $10^3$  y  $10^4$ .

Con objeto de ajustar de forma más precisa el rango de valores de la constante  $\lambda$  se ha realizado una nueva evaluación para un conjunto de valores de  $\lambda$  contenidos dentro

del rango propuesto inicialmente. En concreto esos valores son:  $\lambda = [2 \cdot 10^3, 5 \cdot 10^3, 7 \cdot 10^3, 10^4, 1.5 \cdot 10^4 \text{ y } 2 \cdot 10^4]$ . Los resultados obtenidos se presentan en la Tabla 7.17.

Valor de la constante $\lambda$	Número medio de iteraciones	Error cuadrático medio de la velocidad lineal (mm/s)	Error cuadrático medio de la velocidad angular (grados/s)
$\lambda = 2 \cdot 10^3$	10.04	10.4058	1.6823
$\lambda = 5 \cdot 10^3$	7.84	10.8223	1.6897
$\lambda = 7 \cdot 10^3$	7.04	10.9564	1.6941
$\lambda = 10^4$	6.16	11.1381	1.6959
$\lambda = 1.5 \cdot 10^4$	5.4	11.2665	1.6925
$\lambda = 2 \cdot 10^4$	5.24	11.3235	1.6908

Tabla 7.17. Valores medios de los errores cometido en la estimación de la velocidad lineal y angular y número medio de iteraciones necesarias hasta la convergencia del algoritmo para las 100 imágenes de la secuencia de prueba, en función del valor de la constante  $\lambda$ , para valores de  $\lambda$  entre  $2 \cdot 10^3$  y  $2 \cdot 10^4$ .

Los resultados mostrados en la Tabla 7.17 demuestran que cualquier valor perteneciente al rango definido para  $\lambda$  en este apartado proporciona resultados adecuados respecto al número de iteraciones y el error cuadrático medio de las componentes de velocidad lineal y angular.

La elección del valor de  $\lambda$  a utilizar responde a la búsqueda de una solución de compromiso entre el número de iteraciones y el error cuadrático medio de la velocidad.

### 7.3.3. Validación de los valores de $\lambda$ y $\mu$ elegidos

A lo largo de los apartados 7.3.1 y 7.3.2 se ha realizado un análisis por separado del efecto de cada una de las constantes de ponderación ( $\lambda$  y  $\mu$ ) sobre las diferentes variables implicadas en la función objetivo para  $n_c$  cámaras (ecuación (7.2)) propuesta en esta tesis doctoral.

En este apartado se estudia el efecto que tienen ambas constantes, de forma conjunta, sobre cada uno de los grupos de variables de la función objetivo. En las pruebas experimentales presentadas se considera un conjunto de valores de las constantes de ponderación  $\lambda$  y  $\mu$  incluidos en los rangos determinados en los dos apartados anteriores. Estas pruebas experimentales se realizan tanto para las 100 imágenes de la secuencia 1 (a la que pertenecen las imágenes de la Figura 7.1 y que han sido utilizadas para el ajuste de los valores de las constantes) como con 100 imágenes de las secuencias 2 y 3 (a las que pertenecen las imágenes de la Figura 7.2 (a) y (b)

respectivamente) que también han sido captadas en el espacio inteligente, pero en diferentes condiciones de iluminación.

Los resultados obtenidos para las tres secuencias se presentan a continuación. No se incluye el error cuadrático medio de la velocidad angular debido a que en los apartados anteriores se ha comprobado que la variación de este error es poco significativa frente a la variación del error en la velocidad lineal y el número medio de iteraciones hasta la convergencia.

En la Tabla 7.18 se muestra el valor del error cuadrático medio de la velocidad lineal calculado para 25 imágenes de cada una de las tres secuencias de prueba (a la que pertenecen las imágenes de las Figura 7.1 y Figura 7.2) en función del valor de las constantes de ponderación  $\lambda$  y  $\mu$ . En esta tabla se observa que los resultados obtenidos para la secuencia 1 (Figura 7.1) y la secuencia 3 (Figura 7.2(b)) son similares, mientras que, en el caso de la secuencia 2 (Figura 7.2(a)) el valor del error en la velocidad es incluso más pequeño. En cualquier caso, para los valores de las constantes  $\lambda$  y  $\mu$  dentro de los rangos que han sido determinados en los apartados anteriores, el error cuadrático medio de la velocidad lineal obtenido es inferior a 23 mm/s.

		Error cuadrático medio en la velocidad lineal (mm/s)				
		$\lambda=2 \cdot 10^3$	$\lambda=5 \cdot 10^3$	$\lambda=10^4$	$\lambda=1.5 \cdot 10^4$	$\lambda=2 \cdot 10^4$
Secuencia 1	$\mu = 3 \cdot 10^{-3}$	17.4120	7.1696	7.0773	15.3851	5.9380
	$\mu = 7 \cdot 10^{-3}$	20.3462	8.5796	7.9974	16.0547	7.8087
	$\mu = 10^{-2}$	22.7220	9.6262	9.6665	18.5884	9.5925
	$\mu = 3 \cdot 10^{-2}$	22.8491	9.6082	10.7336	18.4382	10.1995
Secuencia 2	$\mu = 3 \cdot 10^{-3}$	0.0428	0.0190	0.0159	0.0195	0.0072
	$\mu = 7 \cdot 10^{-3}$	0.0428	0.0190	0.0159	0.0195	0.0072
	$\mu = 10^{-2}$	0.0428	0.0190	0.0159	0.0195	0.0072
	$\mu = 3 \cdot 10^{-2}$	0.0428	0.0190	0.0159	0.0195	0.0072
Secuencia 3	$\mu = 3 \cdot 10^{-3}$	11.2763	6.8762	7.4124	5.8075	6.7436
	$\mu = 7 \cdot 10^{-3}$	11.5297	6.8684	7.1848	5.7083	6.8569
	$\mu = 10^{-2}$	11.5573	6.8901	7.1077	5.6515	6.8989
	$\mu = 3 \cdot 10^{-2}$	11.5998	6.8901	7.0328	5.6515	6.8989

Tabla 7.18. Error cuadrático medio de la velocidad lineal (en milímetros/segundo) en función del valor de las constantes  $\lambda$  y  $\mu$ . Los resultados se obtienen para 25 imágenes de las secuencias 1, 2 y 3 (a las que pertenecen las imágenes de la Figura 7.1 y la Figura 7.2).

Por otro lado, en la Tabla 7.19 se presenta el número medio de iteraciones hasta la convergencia. Al igual que en el caso de las velocidades, los resultados obtenidos para la secuencia 1 (Figura 7.1) y la secuencia 3 (Figura 7.2(b)) son similares, siendo en el peor de los casos inferior a 12 iteraciones. En el caso de la secuencia 2 (Figura 7.2(a)) se observa que el número de iteraciones es muy inferior para cualquier valor de  $\mu$  inferior a  $3 \cdot 10^{-2}$ .

Los resultados presentados en la Tabla 7.18 y la Tabla 7.19 permiten concluir que los rangos de valores de las constantes  $\lambda$  y  $\mu$  obtenidos en los apartados anteriores utilizando un conjunto de imágenes de la secuencia 1 son igualmente válidos para las imágenes de las secuencias 2 y 3, a pesar de que dichas secuencias han sido adquiridas en diferentes condiciones de iluminación.

		Numero medio de iteraciones				
		$\lambda=2 \cdot 10^3$	$\lambda=5 \cdot 10^3$	$\lambda=10^4$	$\lambda=1.5 \cdot 10^4$	$\lambda=2 \cdot 10^4$
Secuencia 1	$\mu = 3 \cdot 10^{-3}$	10.76	7.80	5.32	4.12	3.52
	$\mu = 7 \cdot 10^{-3}$	12.76	9.60	6.84	4.84	4.32
	$\mu = 10^{-2}$	12.40	9.32	7.80	6.52	5.76
	$\mu = 3 \cdot 10^{-2}$	10.04	7.84	6.16	5.40	5.24
Secuencia 2	$\mu = 3 \cdot 10^{-3}$	3.75	3.75	3.75	3.75	3.75
	$\mu = 7 \cdot 10^{-3}$	3.75	3.75	3.75	3.75	3.75
	$\mu = 10^{-2}$	3.75	3.75	3.75	3.75	3.75
	$\mu = 3 \cdot 10^{-2}$	3.75	3.75	3.75	3.75	3.75
Secuencia 3	$\mu = 3 \cdot 10^{-3}$	13.00	5.25	6.50	4.25	5.75
	$\mu = 7 \cdot 10^{-3}$	8.75	3.75	4.00	3.50	3.75
	$\mu = 10^{-2}$	8.00	2.50	3.25	2.75	3.00
	$\mu = 3 \cdot 10^{-2}$	7.50	2.50	2.50	2.75	3.00

Tabla 7.19. Número medio de iteraciones en función del valor de las constantes  $\lambda$  y  $\mu$ . Los resultados se obtienen para 25 imágenes de las secuencias 1 (Figura 7.1), 2 (Figura 7.2(a)) y 3 (Figura 7.2(b)).

## 7.4. Conclusiones y aportaciones

En el presente capítulo se han presentado diferentes funciones de preservación de bordes presentes en la literatura. Estas funciones permiten regularizar el término asociado a la profundidad en cualquiera de las dos funciones objetivo, descritas en el capítulo 4. En el estudio realizado se ha corroborado que cualquiera de las funciones de

preservación de bordes presentadas es válida para la aplicación planteada en esta tesis. Esto es así siempre que los valores de las constantes de ponderación de los diferentes términos de la función objetivo se ajusten de forma adecuada.

Para el caso de la función cuadrática (elegida por su simplicidad y porque los resultados no difieren mucho con respecto a otras alternativas) se han obtenido los valores más adecuados para las constantes de ponderación  $\lambda$  y  $\mu$ , en la aplicación de interés. El ajuste de estas constantes ha sido llevado a cabo utilizando un conjunto de imágenes adquiridas en el ISPACE-UAH.

Se parte de que en el capítulo 4 se han presentado dos funciones objetivo diferentes (en función de si incorpora información de una o de múltiples cámaras), realizando un amplio estudio, para cada una de estas funciones, del proceso de ajuste de los valores de  $\lambda$  y  $\mu$ . En ambos casos, los valores elegidos han sido validados con diferentes pruebas experimentales, contemplando diferentes condiciones ambientales y de presencia de robots y otros objetos. En todos los casos se han obtenido resultados satisfactorios.

Entre las aportaciones de este capítulo se destaca el estudio de las diferentes funciones de preservación de bordes que existen en la literatura y que ha permitido concluir que todas ellas son igualmente válidas, siempre que los valores de las constantes se ajusten de forma correcta. El proceso de ajuste de las constantes para la aplicación de interés es otra de las aportaciones realizadas.



## Capítulo 8

# **Identificación y seguimiento de múltiples robots móviles**



El resultado de las etapas de inicialización de variables (curvas y profundidad) y segmentación de movimiento es un conjunto de puntos que pertenecen a cada uno de los robots móviles presentes en la escena.

Tras la minimización de la función objetivo es necesario identificar cuáles de los objetos segmentados son robots móviles y cuáles se corresponden con obstáculos móviles o usuarios. Esta identificación es posible debido a que los robots son agentes controlados por el espacio inteligente, de forma que se dispone de información acerca de los mismos, tanto de los comandos de guiado que se envían a los robots, como de las medidas de los sensores odométricos. Conocido el número de agentes controlados es sencillo realizar su identificación mediante la comparación entre las componentes de velocidad estimadas por el algoritmo de segmentación y posicionamiento para cada uno de los objetos móviles presentes en la escena, y las medidas de los sensores odométricos ubicados a bordo de cada uno de los robots móviles.

Una vez realizada la identificación se incluye también un proceso de seguimiento de los robots móviles que permite realizar una estimación de la posición y las velocidades de dichos robots en la siguiente imagen de la secuencia. Este estimador se incluye con el objetivo de incrementar la robustez del sistema así como reducir el tiempo de cómputo consumido por el proceso de inicialización de variables y minimización de la función objetivo.

A la hora de implementar el seguidor de múltiples objetos es posible encontrar en la literatura diferentes alternativas que pueden dividirse en tres grupos en función de si se usa un estimador por cada uno de los objetos a seguir, se emplea un único estimador aumentando el vector de estado, o bien se usa un único estimador multimodal.

En el caso de esta tesis, inicialmente se planteó el uso de un estimador (en concreto un filtro de Kalman) por cada uno de los robots móviles a seguir. Sin embargo, esta alternativa requiere la inicialización de un nuevo filtro cada vez que un robot entra en la escena. Esto incrementa de forma notable la complejidad de la etapa de seguimiento, así como el tiempo de cómputo consumido en la misma a medida que aumenta el número de robots móviles en la escena. Con objeto de solventar estas deficiencias, en la etapa de seguimiento se ha incluido un filtro de partículas extendido con proceso de clasificación (XPFCP por sus siglas en inglés) [Marron 2008] cuyo carácter multimodal permite realizar el seguimiento de múltiples robots móviles con un único estimador, sin

necesidad de aumentar el vector de estado. Todo ello, cumpliendo los requisitos de robustez y fiabilidad requeridos para el sistema propuesto en esta tesis.

## 8.1. Identificación de los robots móviles

La solución propuesta en esta tesis para la segmentación de movimiento y la posición 3D, proporciona un conjunto de puntos 3D en  $\Gamma_w$  que pertenecen a cada uno de los objetos detectados (que pueden ser tanto robots móviles como usuarios u obstáculos móviles). Así mismo se realiza una estimación de las componentes de velocidades lineal y angular (también respecto a  $\Gamma_w$ ) de dichos objetos.

Con la propuesta realizada, además de identificar los diferentes robots móviles, también es posible diferenciar entre los agentes controlados y los posibles obstáculos y usuarios del espacio inteligente. Esto se lleva a cabo mediante la comparación de las velocidades estimadas, tras la segmentación de movimiento, y las medidas a partir de los sensores odométricos ubicados a bordo de cada robot móvil. Teniendo en cuenta que en el espacio inteligente tanto los robots como los usuarios se desplazan por un plano (y se pueden considerar objetos rígidos), únicamente se comparan las componentes de velocidad lineal en  $X$  e  $Y$ , ya que la velocidad a lo largo del eje  $Z$  es cero en todos los casos.

Partimos de un conjunto de  $n_r$  medidas de velocidad  $\{\mathbf{v}_m^i = (v_{mx}^i, v_{my}^i, v_{mz}^i)\}_{i=1}^{n_r}$  obtenidas de los sensores odométricos de los  $n_r$  robots, y de un conjunto de  $N-1$  velocidades  $\{\mathbf{v}_e^j = (v_{ex}^j, v_{ey}^j, v_{ez}^j)\}_{j=1}^{N-1}$  estimadas para cada uno de los  $N-1$  grupos de puntos 3D  $\mathbf{G}_{j=1:N-1}$  en  $\Gamma_w$ , obtenidos tras la minimización de la función objetivo. Por cada uno de los agentes controlados, Robot $_i$  ( $i=1,2,\dots,n_r$ ), se identifica el grupo de puntos  $\{\mathbf{G}_j\}_{j=1}^{N-1}$  que le corresponde como aquél para el que la diferencia entre las componentes de velocidad lineal medidas por los sensores odométricos del robot  $i$   $\{\mathbf{v}_m^i\}_{i=1}^{n_r}$  y las estimadas  $\{\mathbf{v}_e^j\}_{j=1}^{N-1}$  es mínimo, tal como se expresa en la ecuación (8.1).

$$\mathbf{G}_j \in \text{Robot}_i \Leftrightarrow j = \arg \min_{j=1,2,\dots,N-1} \left\{ \sqrt{(v_{mx}^i - v_{ex}^j)^2 + (v_{my}^i - v_{ey}^j)^2} \right\} \quad i = 1, 2, \dots, n_r \quad (8.1)$$

De esta forma es posible identificar qué grupo de puntos corresponde a cada uno de los robots móviles.

## 8.2. Filtro de partículas extendido con proceso de clasificación (XPFCP)

El XPFCP [Marron 2008] se basa en la combinación de métodos probabilísticos de estimación, y métodos determinísticos de asociación.

El algoritmo probabilístico utilizado como base es un filtro de partículas (PF por sus siglas en inglés) cuyo carácter multimodal permite modelar con una única función la posición de un número variable de robots móviles en cada instante. Para que la creencia  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$  obtenida a la salida del filtro de partículas permita caracterizar un número variable de objetos es necesario incluir una etapa de *re-inicialización* en el modelo básico del filtro de partículas. El filtro de partículas así modificado se conoce como filtro de partículas extendido (XPF) [Koller-Meier 2001]. El XPF es un algoritmo poco robusto, ya que pierde las hipótesis de estimación fácilmente. Con objeto de incrementar la robustez del filtro, en [Marron 2008] se propone la inclusión de un clasificador del conjunto de medidas que se obtienen en cada instante, que permite realizar el seguimiento de un número variable de robots de forma robusta. Por otro lado, la salida determinística se obtiene aplicando un nuevo clasificador a las partículas obtenidas a la salida del filtro. La inclusión de estos dos clasificadores da lugar al XPFCP que será el utilizado en esta tesis para el seguimiento de un número variable de robots móviles.

En el diagrama de bloques de la Figura 8.1 se presentan las diferentes etapas que conforman el XPFCP. En esta figura, los bloques correspondientes al XPF se han representado en color azul, mientras que las etapas de clasificación de las medidas y las partículas se muestran en color naranja.

En los siguientes apartados se describe brevemente cada uno de los algoritmos que conforman el XPFCP. En primer lugar, en el apartado 8.2.1 se presenta el filtro de partículas extendido, destacando las modificaciones que se introducen por el hecho de incluir un proceso de clasificación de medidas para, posteriormente en el apartado 8.2.2 describir el algoritmo de clasificación utilizado.

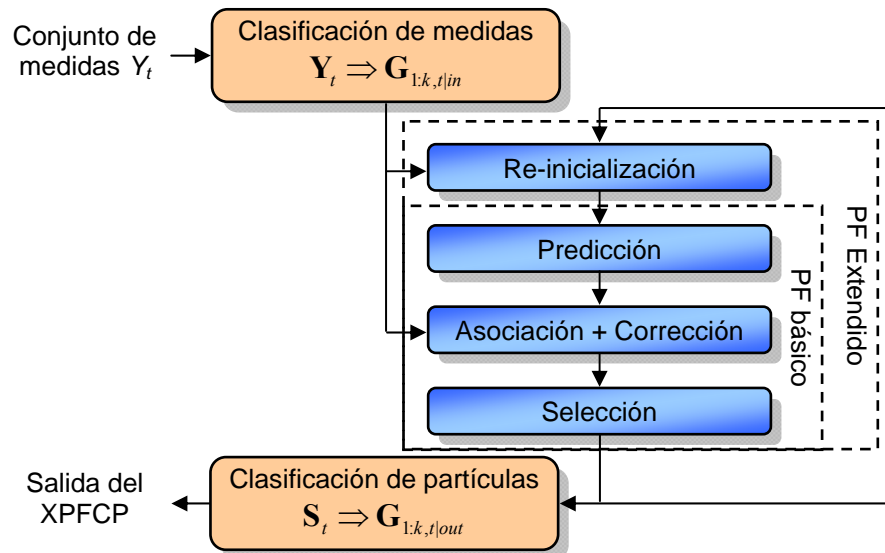


Figura 8.1. Diagrama de bloques general del XPFCP propuesto en [Marron 2008]. Las etapas correspondientes al filtro de partículas extendido se muestran en color azul, mientras que las etapas de clasificación de medidas y de partículas se han representado en color naranja.

### 8.2.1. Filtro de partículas extendido (XPF)

El filtro de partículas (PF) es un estimador recursivo probabilístico que basa su funcionamiento en la representación discreta de la creencia, por medio de un conjunto  $S_t$  de  $n_p$  muestras, denominadas partículas ( $S_t = \{\mathbf{s}_t^{(i)} = \mathbf{x}_t^{(i)}\}_{i=1}^{n_p}$ ), que son copias del vector de estado,  $\mathbf{x}_t$ , del sistema a estimar. Cada una de estas muestras está, además, ponderada por un peso normalizado  $\omega_t = \{\tilde{\omega}_t^{(i)}\}_{i=1}^{n_p}$  que caracteriza su probabilidad de certidumbre dentro de la distribución de probabilidad global. De esta forma se obtiene la creencia discreta (8.2), que es la salida del filtro de partículas, a partir de la cual se extrae el valor final de la estimación.

$$p(\mathbf{x}_t^{(1:n_p)} | \mathbf{y}_{1:t}) \cong S_t = \{\mathbf{x}_t^{(i)}, \tilde{\omega}_t^{(i)}\}_{i=1}^{n_p} \quad (8.2)$$

El PF es una particularización del filtro de Bayes, en la que se discretiza la creencia. Esto permite la implementación del filtro en tiempo real sin discretizar el espacio de estado, o linealizar el modelo del sistema. De este modo, es posible obtener resultados de estimación más precisos que con otros algoritmos como son el filtro de Kalman, o los algoritmos basados en rejilla.

El PF empleado en el XPFCP incluye una etapa de reinicialización, además de las etapas de predicción, corrección y selección del filtro de partículas básico. Cada una de estas etapas se describe brevemente a continuación.

### Reinicialización:

La inclusión de esta etapa permite la incorporación de nuevas hipótesis de seguimiento en la creencia discretizada del PF. Esto se realiza mediante la incorporación de un nuevo conjunto de partículas al conjunto  $S_{t-1} \cong p(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1})$  que caracteriza la probabilidad a posteriori obtenida en el instante temporal anterior, t-1. De esta forma se genera un nuevo conjunto de partículas  $\hat{S}_{t-1}$ .

Para que la nueva función densidad de probabilidad  $\hat{p}(\mathbf{x}_{t-1} | \mathbf{y}_{t-1})$  represente de forma robusta las distintas hipótesis, las nuevas partículas se obtienen a partir del resultado de la clasificación de medidas en ese instante temporal ( $Y_{t-1} \Rightarrow G_{1:k,t-1|in}$ ). Esto permite incrementar la robustez del filtro, evitando el empobrecimiento del conjunto de partículas.

Para que el número de partículas del filtro ( $n_p$ ) se mantenga constante, el número de partículas que se añaden en esta etapa, coincide con el número que se elimina en la etapa de selección ( $n_{m,t-1}$ ). Denominando  $\gamma_{p,t}$  a la relación  $\gamma_{p,t} = \frac{n_{m,t-1}}{n_p}$ , la función densidad de probabilidad  $\hat{p}(\mathbf{x}_{t-1} | \mathbf{y}_{t-1})$  del filtro de partículas puede expresarse mediante la ecuación (8.3).

$$\hat{p}(\mathbf{x}_{t-1} | \mathbf{y}_{t-1}) = \gamma_{p,t} \cdot p(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) + (1 - \gamma_{p,t}) \cdot p(G_{1:k,t-1|in}) \quad (8.3)$$

### Predicción:

Durante la etapa de predicción del PF, cada una de las partículas procedentes del paso de reinicialización  $\hat{S}_{t-1} = \left\{ \hat{\mathbf{x}}_{t-1}^{(i)}, \frac{1}{n} \right\}_{i=1}^{n_p} \cong \hat{p}(\mathbf{x}_{t-1}^{(1:n_p)} | \mathbf{y}_{1:t-1})$  se propaga al siguiente instante temporal mediante el modelo de estado  $p(\mathbf{x}_t | \mathbf{x}_{t-1})$ , aplicando la ecuación (8.4). De esta forma se obtiene el valor a priori de la creencia  $S_{t|t-1} = \left\{ \mathbf{x}_{t|t-1}^{(i)}, \tilde{\omega}_{t-1} \right\}_{i=1}^{n_p} \cong \hat{p}(\mathbf{x}_t^{(1:n_p)} | \mathbf{y}_{1:t-1})$ , representado mediante un conjunto de muestras, con sus correspondientes pesos.

$$\mathbf{x}_{t|t-1}^{(1:n_p)} = \mathbf{x}_{t-1}^{(1:n_p)} \cdot p(\mathbf{x}_t | \mathbf{x}_{t-1}) \quad (8.4)$$

**Corrección:**

En el paso de corrección se obtiene la función de probabilidad a posteriori  $p(\mathbf{x}_t^{(1:n_p)} | \mathbf{y}_{1:t})$  a partir de la creencia muestreada  $p(\mathbf{x}_t^{(1:n_p)} | \mathbf{y}_{1:t-1})$  calculada en la etapa de predicción, mediante el muestreo de Monte-Carlo (ecuación (8.5)).

$$p(\mathbf{x}_t^{(1:n_p)} | \mathbf{y}_{1:t}) = \sum_{i=1}^{n_p} \tilde{\omega}_t^{(i)} \cdot \delta(\mathbf{x}_t^{(i)}) \quad (8.5)$$

El cálculo de los pesos normalizados de las muestras  $\boldsymbol{\omega}_t = \{\omega_t^{(i)}\}_{i=1}^{n_p}$  que caracterizan a la creencia  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$  se realiza, de forma recursiva, a través de las expresiones (8.6) y (8.7).

$$\omega(\mathbf{x}_{0:t}) = \omega(\mathbf{x}_{0:t-1}) \cdot \frac{p(\mathbf{y}_t | \mathbf{x}_t) \cdot p(\mathbf{x}_t | \mathbf{x}_{t-1})}{q(\mathbf{x}_t | \mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})} \quad (8.6)$$

$$\tilde{\omega}_t^{(i)} = \frac{\omega_t^{(i)}}{\sum_{i=1}^{n_p} \omega_t^{(i)}} \quad (8.7)$$

donde  $q(\mathbf{x}_t | \mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$  es la función de aproximación a la creencia. En este trabajo, dicha función se sustituye por el modelo de actuación del sistema, reduciéndose la ecuación (8.6) a la expresión mostrada en la ecuación (8.8).

$$\omega(\mathbf{x}_{0:t}) = \omega(\mathbf{x}_{0:t-1}) \cdot p(\mathbf{y}_t | \mathbf{x}_t) \Rightarrow \omega_t^{1:n_p} = \omega_{t-1}^{1:n_p} \cdot p(\mathbf{y}_t | \mathbf{x}_t^{1:n_p}) \quad (8.8)$$

En [Koller-Meier 2001], la función de verosimilitud  $p(\mathbf{y}_t | \mathbf{x}_t) \cong p(\mathbf{y}_t | \mathbf{x}_t^{1:n_p})$  usada para cada partícula, es una función gaussiana cuya media es la proyección de cada partícula en el espacio de medidas definido para la aplicación de interés, y cuya matriz de covarianza viene dada por el modelo de observación del sistema.

En el caso del XPFCP propuesto en [Marron 2008], la función de verosimilitud considerada en la etapa de corrección se genera a partir de los centroides  $\mathbf{g}_{1:k,t|in}$  obtenidos a la salida del clasificador de medidas. Esto es así porque estos centroides representan de forma más robusta al conjunto de medidas, quedando la expresión (8.8), como se muestra en la ecuación (8.9).

$$\omega_t^{1:n_p} = \omega_{t-1}^{1:n_p} \cdot p(\mathbf{g}_{1:k,t|in} | \mathbf{x}_t) \quad (8.9)$$

El valor discreto de cada peso  $\tilde{\omega}_t^{(i)}$  se obtiene buscando el máximo de la gaussiana entre las realizaciones que se obtienen de la misma para cada una de las medidas. Dicho



máximo viene dado por la distancia mínima entre la media de la gaussiana y la medida más próxima a ésta, según se muestra en la ecuación (8.10)

$$\omega_t^{(i)} = \omega_{t-1}^{(i)} \cdot e^{-d_{\min,t,t}^2/2O} \quad (8.10)$$

Siendo:  $d_{\min,t,t}$  el valor mínimo de la distancia entre cada la proyección de la partícula  $\mathbf{x}_t^{(i)}$  en el espacio de medidas y cada una de las medidas  $\{\mathbf{y}_t^{(j)}\}_{j=1}^{m_t}$  en el instante  $t$ , y siendo  $O$  la varianza de la gaussiana que coincide con la covarianza del modelo de observación.

Tras obtener todos los pesos, estos se normalizan empleando la expresión mostrada en la ecuación (8.7).

De esta forma, tras las etapas de predicción, y corrección, se obtiene el conjunto de partículas  $S_t = \{\mathbf{x}_t^{(i)}, \tilde{\omega}_t^{(i)}\}_{i=1}^{n_p}$  que describe la creencia  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$ . El objetivo de la etapa de corrección es la obtención de un peso  $\tilde{\omega}_t^{(i)}$  para cada partícula  $\mathbf{x}_t^{(i)}$  que pondere la probabilidad de acierto de dicha partícula en el proceso de estimación.

### Selección:

Esta etapa se ejecuta tras los pasos de predicción y corrección, y en ella se seleccionan un conjunto de  $n_p - n_{m,t}$  partículas, del conjunto de muestras ponderadas de la creencia  $S_t = \{\mathbf{x}_t^{(i)}, \tilde{\omega}_t^{(i)}\}_{i=1}^{n_p}$ , en función de su peso. El paso de reinicialización descrito anteriormente se ejecuta en el instante temporal  $t+1$ , sobre las partículas seleccionadas en  $t$ .

## 8.2.2. Clasificador

Tal como se puede observar en el diagrama de bloques mostrado en la Figura 8.1, el XPFCP propuesto en [Marron 2008] incluye dos clasificadores diferentes. El primero se ejecuta a la entrada del filtro agrupando las medidas, mientras que el segundo se encarga de clasificar las partículas, haciendo posible que la salida del XPFCP sea determinística.

Respecto a los algoritmos de clasificación, en [Marron 2008] se presentan dos alternativas, una de ellas basada en el estándar k-medias, y la otra basada en el algoritmo borroso *subtractive*, comprobándose que con ambas técnicas se obtienen

resultados similares. En el caso de esta tesis se ha decidido emplear el algoritmo  $k$ -medias, ya que a pesar de su sencillez, proporciona buenos resultados en las condiciones planteadas en la tesis.

Dado que la técnica  $k$ -medias ha sido utilizada previamente en la etapa de inicialización de variables, tanto la nomenclatura utilizada como los fundamentos teóricos de esta técnica han sido expuestos con detalle a lo largo del apartado 6.2.1, por lo que no se considera necesario incluirlo de nuevo en este capítulo.

Cabe destacar que en el XPFCP utilizado para la propuesta de seguimiento de múltiples robots móviles que se hace en esta tesis no se incluye el clasificador de medidas. Ello es debido a que a la salida de la etapa de segmentación y posicionamiento 3D de los robots se cuenta con un conjunto de medidas ya clasificadas, que se introducen de forma directa a la etapa de re-inicialización, tal como se expondrá con mayor detalle en el siguiente apartado.

### 8.3. XPFCP para el seguimiento de múltiples robots móviles

El objetivo de la etapa de seguimiento basado en el uso del XPFCP es la predicción de la posición y las velocidades de los robots móviles en la siguiente imagen de la secuencia. Para ello parte de los resultados obtenidos tras la segmentación de movimiento y la identificación de los agentes controlados. En la Figura 8.2 se muestra un diagrama de bloques en el que se han incluido las diferentes etapas que conforman la solución propuesta para segmentación y posicionamiento de robots móviles, incluyendo la identificación y el seguimiento de los mismos.

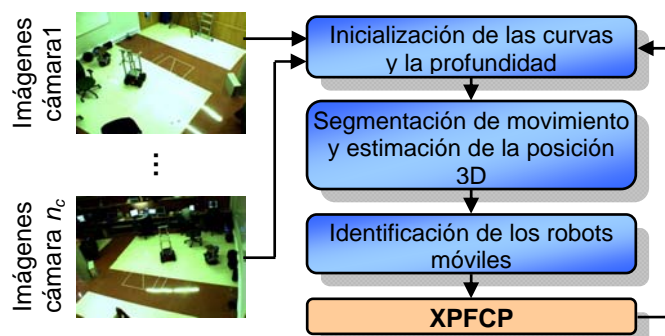


Figura 8.2. Diagrama de bloques general de la solución propuesta para la segmentación de movimiento y la estimación de la posición 3D de un número variable de robots móviles.

La predicción obtenida a la salida del XPFCP se realimenta a las etapas anteriores según se expone en los siguientes apartados.

### 8.3.1. Modelo del sistema

Para caracterizar la evolución de la posición y la velocidad de cada uno de los robots móviles a seguir, se plantea un modelo discreto, genérico y de primer orden a partir de los valores de las coordenadas que definen la posición de los robots en cada instante, mediante las expresiones mostradas en la ecuación (8.11).

$$\begin{aligned}x_{t+1} &= x_t + v_{x,t} \cdot t_s \\y_{t+1} &= y_t + v_{y,t} \cdot t_s \\z_{t+1} &= z_t\end{aligned}\tag{8.11}$$

Se definen los vectores de estado  $\mathbf{x}_t$  y de medida  $\mathbf{y}_t$  en el instante temporal  $t$  mediante los vectores de dimensión  $5 \times 1$  mostrados en las ecuaciones (8.12) y (8.13):

$$\mathbf{x}_t = \begin{bmatrix} X_{w,t} & Y_{w,t} & Z_{w,t} & v_{x,t}^w & v_{y,t}^w \end{bmatrix}^T\tag{8.12}$$

$$\mathbf{y}_t = \begin{bmatrix} X_{w,t} & Y_{w,t} & Z_{w,t} & v_{x,t}^w & v_{y,t}^w \end{bmatrix}^T\tag{8.13}$$

Estas ecuaciones del modelo del sistema, se pueden expresar en variables de estado según se muestra en las ecuaciones (8.14) y (8.15).

$$\mathbf{x}_{t+1} = \mathbf{A} \cdot \mathbf{x}_t\tag{8.14}$$

$$\mathbf{y}_t = \mathbf{C} \cdot \mathbf{x}_t\tag{8.15}$$

siendo A y C las siguientes matrices, en las que  $t_s$  representa el tiempo de muestreo.

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & t_s & 0 \\ 0 & 1 & 0 & 0 & t_s \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ y } \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Cabe destacar que, en este caso, es posible incluir las componentes de velocidad lineal en  $X_w$  e  $Y_w$  en el vector de medida  $\mathbf{y}_t$ . Ello es debido a que la solución propuesta

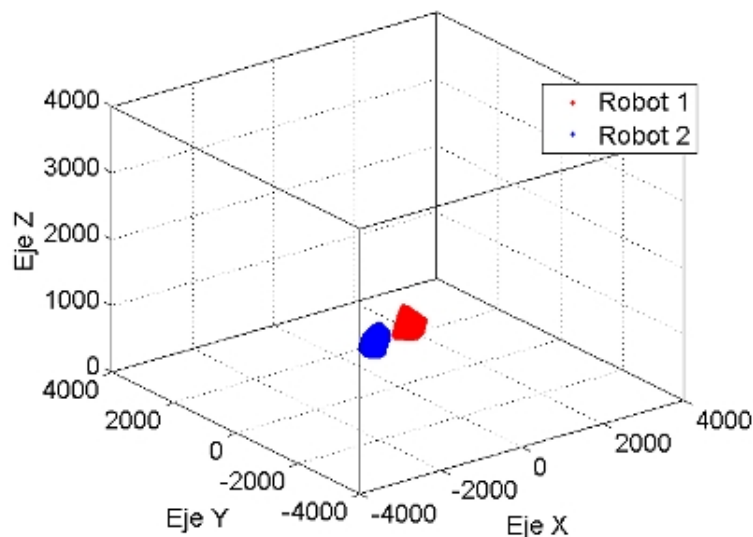
para la segmentación de movimiento y posicionamiento 3D, descrita en capítulos anteriores, permite obtener una estimación de la velocidad de los robots móviles.

### 8.3.2. Seguimiento mediante XPFCP

Dado un conjunto de imágenes de entrada adquiridas por  $n_c$  cámaras fijas y calibradas, tras la inicialización de variables, la segmentación de movimiento y la estimación de la posición 3D de los robots móviles, se tiene un conjunto de puntos en 3D en  $\Gamma_w$  que definen los contornos exteriores de los robots tal como se muestra en el ejemplo de la Figura 8.3.



(a)



(b)

Figura 8.3. (a) Imágenes de entrada pertenecientes a la secuencia de pruebas vistas por tres cámaras fijas en el ISPACE-UAH. (b) Conjunto de puntos obtenidos tras la minimización de la función objetivo y la identificación de los robots móviles contenidos en la escena mostrada en las imágenes anteriores.

Además, se cuenta con una estimación de las componentes de velocidad lineal y angular de cada uno de los grupos de puntos  $\{G_j\}_{j=1}^{N-1}$ . A partir de esta información se obtienen los vectores de medida que se incorporarán al XPFCP.

A continuación se describe brevemente la implementación de cada una de las etapas del XPFCP para la solución planteada en esta tesis.

### 8.3.2.1. Implementación del XPFCP

#### Reinicialización:

Por cada una de las medidas en 3D obtenidas a partir de la etapa de segmentación y posicionamiento de los robots móviles, se genera un vector formado por las coordenadas 3D del punto en  $\Gamma_w$ , y las componentes de velocidad lineal en  $X_w$  e  $Y_w$  del objeto  $\{G_j\}_{j=1}^{N-1}$  al que pertenece dicho punto (ecuación (8.13)).

Estos vectores son los que se introducen a la etapa de reinicialización (o de inicialización, en el caso de la primera iteración) del XPF. De ellos se obtienen las  $n_{m,t-1}$  partículas que se incorporan a las resultantes de la ejecución anterior del filtro de partículas  $S_{t-1} \cong p(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1})$  que caracteriza la probabilidad a posteriori obtenida en el instante temporal anterior, t-1. De esta forma se genera un nuevo conjunto de partículas  $\hat{S}_{t-1}$ . El número de partículas a incorporar en la re-inicialización ( $n_{m,t-1}$ ) es el mismo que se elimina en la etapa de selección de la iteración anterior del filtro, tal como se ha indicado anteriormente.

#### Predicción:

Durante la etapa de predicción del PF, cada una de las partículas procedentes del paso de reinicialización  $\hat{S}_{t-1} = \left\{ \hat{\mathbf{x}}_{t-1}^{(i)}, \frac{1}{n} \right\}_{i=1}^{n_p} \cong \hat{p}(\mathbf{x}_{t-1}^{(1:n_p)} | \mathbf{y}_{1:t-1})$  se propaga al siguiente instante temporal, mediante el modelo de estado definido en la ecuación (8.14). De esta forma se obtiene el valor a priori de la creencia  $S_{t|t-1} = \left\{ \mathbf{x}_{t|t-1}^{(i)}, \tilde{\omega}_{t-1} \right\}_{i=1}^{n_p} \cong \hat{p}(\mathbf{x}_t^{(1:n_p)} | \mathbf{y}_{1:t-1})$ , representado mediante un conjunto de muestras, con sus correspondientes pesos.

#### Corrección:

En el paso de corrección se obtiene la función de probabilidad a posteriori  $p(\mathbf{x}_t^{(1:n_p)} | \mathbf{y}_{1:t})$  a partir de la creencia muestreada  $p(\mathbf{x}_t^{(1:n_p)} | \mathbf{y}_{1:t-1})$ , calculada en la etapa de predicción. El cálculo de los pesos normalizados de las muestras  $\omega_t = \left\{ \omega_t^{(i)} \right\}_{i=1}^{n_p}$  que caracterizan a la creencia  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$  se realiza, de forma recursiva, a través de las expresiones (8.6) y (8.7).

Al igual que en [Marron 2008], en esta tesis la función de verosimilitud considerada en la etapa de corrección se genera a partir de los centroides  $\mathbf{g}_{l,k,t/in}$  de los grupos  $\{\mathbf{G}_j\}_{j=1}^{N-1}$ , obtenidos a la salida de las etapas de segmentación y posicionamiento. El valor discreto de cada peso  $\tilde{\omega}_t^{(i)}$  se obtiene buscando el máximo de la gaussiana entre las realizaciones que se obtienen de la misma, para cada una de las medidas. Dicho máximo viene dado por la distancia mínima entre la media de la gaussiana y la medida más próxima a ésta (ecuación (8.10)).

Tras normalizar los pesos, se obtiene el conjunto de partículas  $S_t = \{\mathbf{x}_t^{(i)}, \tilde{\omega}_t^{(i)}\}_{i=1}^{n_p}$  que describe la creencia  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$ . El objetivo de la etapa de corrección es la obtención de un peso  $\tilde{\omega}_t^{(i)}$  para cada partícula  $\mathbf{x}_t^{(i)}$  que pondere la probabilidad de acierto de dicha partícula en el proceso de estimación.

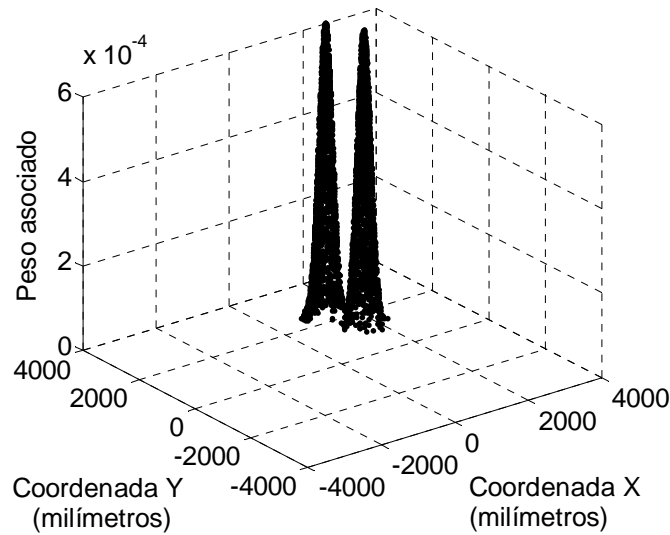


Figura 8.4. Representación gráfica del peso normalizado asociado a cada una de las partículas en el paso de corrección del XPF. Las coordenadas X e Y coinciden con las coordenadas de la partícula en  $\Gamma_w$ .

### Selección:

Esta etapa se ejecuta tras los pasos de predicción y corrección del estimador, y en ella se seleccionan un conjunto de  $n_p - n_{m,t}$  partículas, del conjunto de muestras ponderadas de la creencia  $S'_t = \{\mathbf{x}_t^{(i)}, \tilde{\omega}_t^{(i)}\}_{i=1}^{n_p}$ , en función de su peso.

#### 8.3.2.2. Clasificador

El proceso de clasificación de las partículas añadido al XPF, en el XPFCP, permite obtener una salida determinística a partir de la creencia discreta que genera el filtro.

Para la clasificación se ha empleado la técnica k-medias que ya ha sido descrita en el apartado 6.2 de esta tesis. En el apartado 6.2 se define también el espacio de características empleado para la clasificación (coordenadas  $X_w$  e  $Y_w$  de las medidas) y el valor de distancia  $dMax$  que determina la creación de una nueva clase para la clasificación de las medidas.

Partiendo de que el vector de estado que define las partículas está formado por la posición 3D y las componentes de velocidad lineal de las mismas, para la clasificación se emplea el mismo espacio de características que en el caso de las medidas. Es decir, el clasificador basado en k-medias se aplica sobre las coordenadas de las partículas en el plano  $X_w Y_w$ . Esto implica que el estudio realizado en el capítulo 6, para la clasificación de las medidas, es igualmente válido para el caso de las partículas.

En la Figura 8.5 se muestra, a modo de ejemplo, la salida del XPFCP para el conjunto de partículas representadas con sus pesos correspondientes en la Figura 8.4.

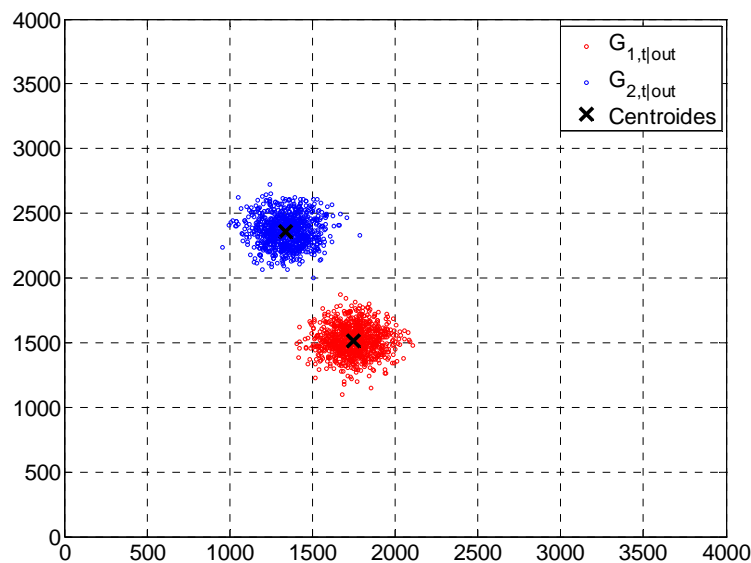


Figura 8.5. Representación sobre el plano  $X_w Y_w$  del resultado de la clasificación del conjunto de partículas mostrado en la Figura 8.4.

### 8.3.3. Realimentación de la salida del XPFCP al módulo de inicialización de variables

El resultado de la clasificación de partículas en el XPFCP proporciona una buena estimación de la posición que ocupará cada uno de los robots en un instante posterior. Esta información se incorpora a la inicialización de variables, tal como se muestra en la Figura 8.2. En concreto, la información de salida del XPFCP se realimenta a las etapas

de obtención de la rejilla de ocupación, mediante *Visual Hull 3D (VH3D)*, y de estimación del número de robots móviles, mediante k-medias.

La obtención de la rejilla de ocupación 3D, usando VH3D, a partir de las curvas iniciales, es una de las tareas con mayor coste computacional de la etapa de inicialización de variables de la solución propuesta.

Partiendo de que el tiempo consumido en la obtención del VH3D se incrementa a medida que el tamaño ( $\Delta h$ ) de los cubos en que se divide el espacio 3D es menor, se ha decidido utilizar la predicción del XPFCP (obtenida para la imagen anterior) para determinar en que zonas de las escena es más probable que se encuentren los robots móviles.

En este punto puede plantearse la alternativa de ejecutar la inicialización únicamente en el área de mayor probabilidad, sin embargo, es importante realizar la búsqueda en toda la imagen para detectar la entrada de nuevos robots o usuarios en la escena. Por este motivo, se ha decidido aplicar VH3D de forma jerárquica. Para ello se ejecuta VH3D con una resolución superior en el área donde la probabilidad de encontrar robots móviles es mayor, y con una resolución más pequeña en el resto de la escena.

Para realizar esto, en primer lugar se define un cubo en 3D alrededor de cada uno de los grupos de partículas obtenidos a la salida del XPFCP. Las dimensiones del cubo en  $X_w$  e  $Y_w$  se establecen de forma que todas las partículas del grupo se encuentren dentro del mismo. Respecto a la dimensión en el eje Z, ésta se fija a 1 metro. Un ejemplo se presenta en la Figura 8.6.

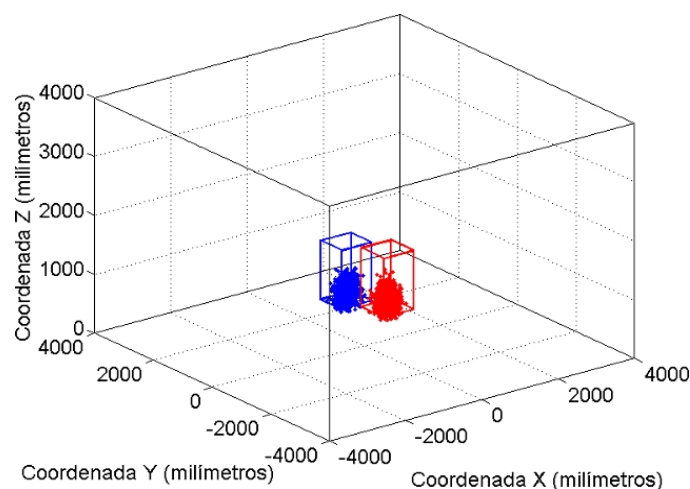


Figura 8.6. Cubos obtenidos a partir de los grupos de partículas a la salida del XPFCP. El interior de estos cubos define la zona donde existe una mayor probabilidad de encontrar robots móviles en la siguiente imagen de la secuencia.



Los cubos definidos alrededor de los grupos de partículas se proyectan sobre el plano imagen de cada una de las cámaras usando las matrices de transformación  $\{\mathbf{R}_{wc}^i, \mathbf{t}_{wc}^i\}_{i=1}^{n_c}$ . De este modo se obtiene una máscara por cada cámara, a la que denominamos  $\{M_{PF}^i\}_{i=1}^{n_c}$ . Esta máscara se aplica a cada una de las imágenes de entrada, separando las curvas iniciales que tienen mayor probabilidad de definir los contornos de los robots móviles del resto de curvas. El área definida por los cubos de la Figura 8.6 sobre el plano imagen de cada cámara puede observarse en el ejemplo mostrado en la Figura 8.7.

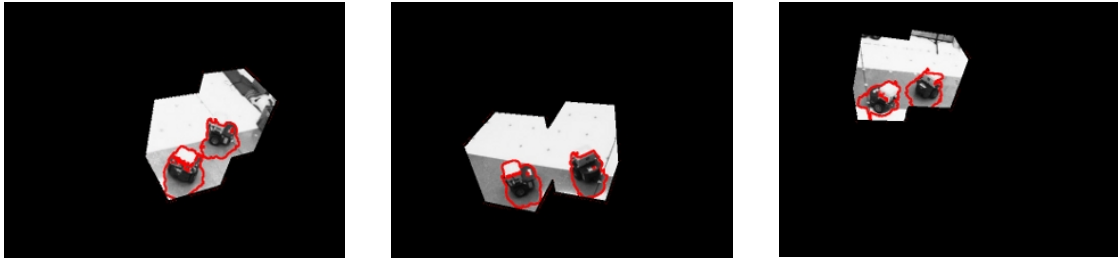


Figura 8.7. Resultado de la proyección de los cubos definidos alrededor de cada grupo de partículas sobre el plano imagen de cada una de las tres cámaras.

Tras la separación de las curvas, se obtiene el VH3D. Esto se hace en dos etapas diferentes. En primer lugar, se aplica VH3D sobre las curvas iniciales con mayor probabilidad de pertenecer a un robot móvil con un tamaño de cubo menor (mayor resolución). A continuación se ejecuta de nuevo VH3D sobre el resto de curvas de la imagen con una resolución menor. Así se obtiene una rejilla de ocupación con mayor densidad de puntos en la zona en que la probabilidad de que se encuentren los robots móviles es más alta. En el caso de esta tesis se ha decidido utilizar un tamaño de rejilla con  $\Delta h = 10cm$  para la zona en que existe mayor probabilidad de que se encuentren los robots móviles y una rejilla con  $\Delta h = 30cm$  para el resto de la escena. En la elección de estos valores se ha tratado de alcanzar una solución de compromiso entre la calidad de los resultados y el tiempo de procesamiento.

Las diferentes etapas para la obtención de la rejilla de ocupación se muestran en el diagrama de bloques de la Figura 8.8.

Además de en la obtención del VH3D, la información dada por el XPFCP también se incorpora en el clasificador incluido en la inicialización de variables. En concreto, los centroides de los grupos de partículas se emplean como valor inicial de los centroides de las diferentes clases en el algoritmo de clasificación (basado en k-medias), que se ejecuta sobre la rejilla de ocupación 3D dada por VH.

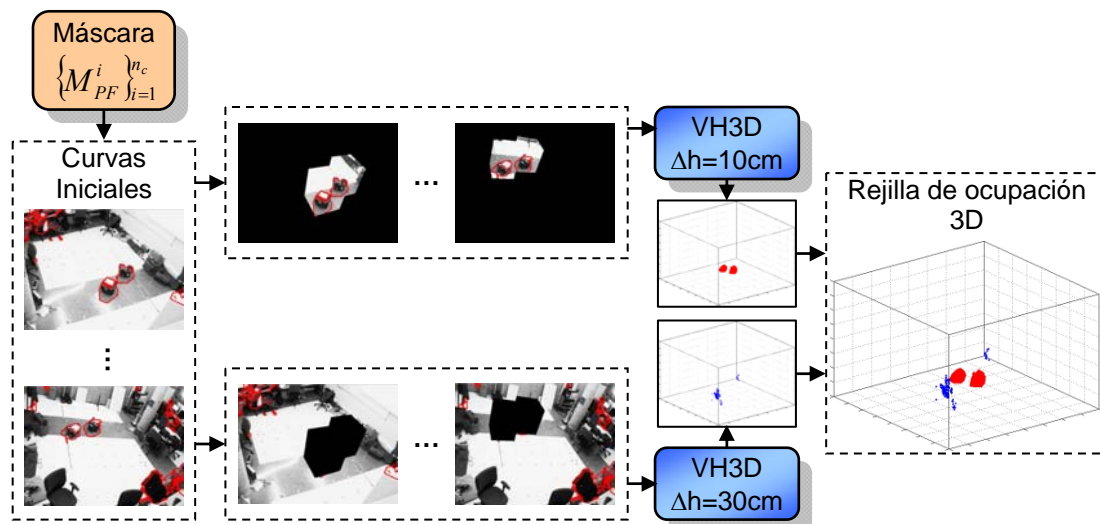


Figura 8.8. Diagrama de bloques general del proceso de obtención del VH3D, utilizando una máscara, a partir de las curvas iniciales que incorpora la información obtenida del XPFCP en la imagen anterior.

En caso de que la predicción del XPFCP sea correcta, el uso de los centroides de los grupos de partículas para la inicialización de los *clusters*, en la clasificación mediante *k*-medias, permite reducir el número de iteraciones hasta la convergencia, con la consiguiente reducción del tiempo de procesamiento.

## 8.4. Conclusiones y aportaciones

En este capítulo se han descrito las etapas de identificación y seguimiento que forman parte de la solución propuesta en esta tesis para la segmentación y posicionamiento de múltiples robots móviles.

La identificación de los robots se lleva a cabo mediante la comparación de las componentes de la velocidad lineal en  $X_w$  e  $Y_w$  estimadas por el algoritmo de segmentación de movimiento y las medidas de los sensores odométricos a bordo de los mismos. Esto es posible porque los robots son agentes controlados por el sistema de supervisión del espacio inteligente. Además de identificar los diferentes robots, en esta etapa, también es posible diferenciar los robots de los usuarios u obstáculos presentes en la escena.

Para el seguimiento de los robots se emplea un filtro de partículas extendido con proceso de clasificación (XPFCP). El XPFCP permite realizar el seguimiento de un número variable de robots con un único estimador, sin necesidad de aumentar el vector de estado.

La información obtenida a la salida del XPFCP se realimenta a la etapa de inicialización de variables, ya que supone una buena estimación de la posición que ocuparán los robots móviles en la siguiente imagen de la secuencia. Esta propuesta de realimentación es la principal aportación de este capítulo. La importancia de la misma radica en que permite reducir el tiempo consumido en la inicialización. La validez de esta propuesta queda demostrada mediante pruebas experimentales realizadas en el ISPACE-UAH tal como se expone en el capítulo 10 de esta memoria.



## Capítulo 9

# **Resultados y validación de la solución propuesta**



En este capítulo se presentan los resultados de las diferentes pruebas experimentales realizadas para la validación empírica de las diferentes etapas que conforman la solución propuesta en esta tesis.

Dada la diversidad de las propuestas realizadas a lo largo de los diferentes capítulos de esta tesis, los resultados presentados en este capítulo se han estructurado en función de la etapa que se evalúa en cada caso. Por ello, la distribución de este capítulo es la siguiente:

- Se comienza validando las diferentes propuestas realizadas en relación con la inicialización de variables (tanto de las curvas como de la profundidad).
- A continuación, se presentan los resultados obtenidos para las dos alternativas planteadas en la segmentación de movimiento y la estimación de la posición 3D de los robots móviles.
- La última etapa analizada es la relacionada con la identificación y el seguimiento de los robots móviles mediante un XPFCP.
- Para finalizar, se presentan los resultados obtenidos al utilizar la solución global propuesta para la segmentación de movimiento y la estimación de la posición 3D de múltiples robots móviles. Con ello se pretende validar la propuesta realizada en esta tesis en su conjunto, pensando en el objetivo final perseguido.

En algunas de las etapas se muestran los valores del tiempo consumido por los diferentes módulos de la solución propuesta. Estos valores han sido obtenidos ejecutando los algoritmos descritos en Matlab. El PC utilizado es un Intel Core 2 CPU 6600 a 2.4GHz, con 3.5GB de RAM.

## **9.1. Descripción de las secuencias de imágenes utilizadas en la evaluación**

Para la validación empírica de la solución propuesta en esta tesis en referencia a la segmentación y posicionamiento de múltiples robots móviles en un espacio inteligente, se han realizado diversas pruebas experimentales. En estas pruebas se han utilizado diferentes secuencias de imágenes. Cabe destacar que todas las secuencias de imágenes

consideradas han sido adquiridas por tres o cuatro cámaras fijas, calibradas y sincronizadas entre sí en el ISPACE-UAH. Las imágenes han sido adquiridas a 15 fps en formato RGB, y con una resolución espacial de 640x480 píxeles.

Además de las imágenes en las que aparecen robots móviles o usuarios, en todas las secuencias se cuenta con un conjunto de imágenes del fondo de la escena, a partir de las que se realiza el modelado del mismo.

Por otro lado, se dispone de la información proporcionada por los sensores odométricos de los robots móviles.

Las características más destacadas de las secuencias empleadas para la validación de las propuestas realizadas, así como algunas imágenes pertenecientes a cada una de ellas se presentan a continuación.

Las imágenes pertenecientes a la secuencia 1 han sido adquiridas por cuatro cámaras. A lo largo de toda la secuencia, un robot móvil se desplaza a través del área de interés. En la Figura 9.1 se muestran dos escenas, (a) y (b), pertenecientes a esta secuencia, en las que puede observarse un único robot móvil en diferentes posiciones. En esta figura únicamente se muestran tres de las cuatro vistas disponibles.

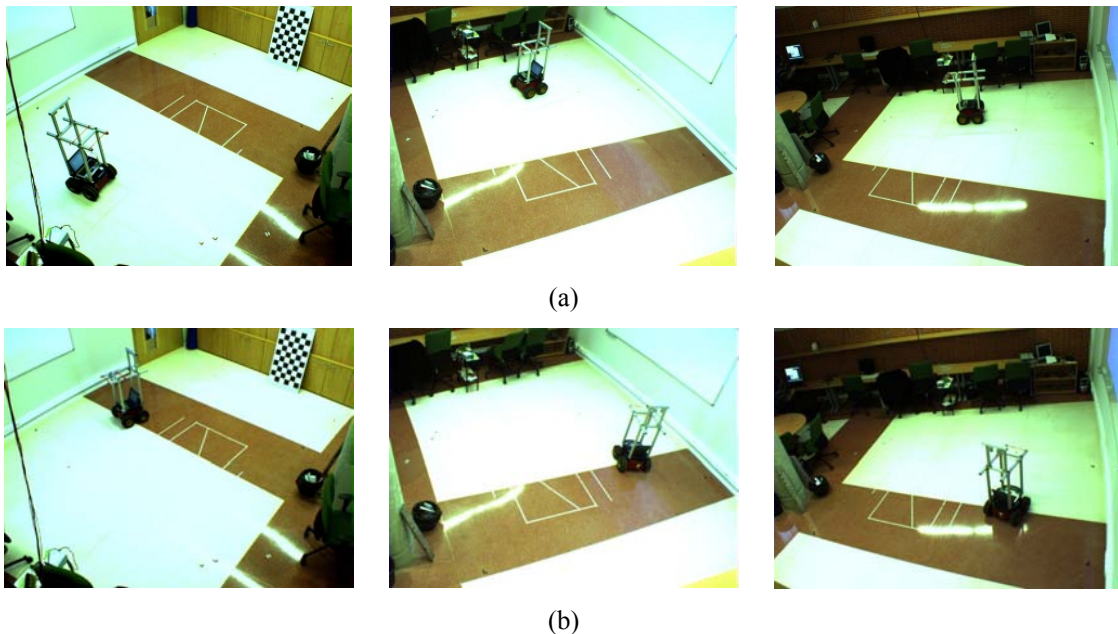


Figura 9.1. Dos ejemplos (a) y (b) de escenas pertenecientes a la secuencia 1 adquiridas por cuatro cámaras en el ISPACE-UAH, de las cuales solamente se representa la información captada por tres cámaras.



Cabe destacar que el ajuste de los diferentes parámetros implicados en la solución propuesta (constantes de ponderación  $\lambda$  y  $\mu$ , umbral dinámico, etc.) ha sido realizado utilizando las imágenes de la secuencia 1. Los valores obtenidos a partir de esta secuencia se utilizan en todas las demás, permitiendo verificar que los valores elegidos para dichos parámetros son válidos a pesar de que cambien las condiciones de iluminación la escena.

La secuencia 2 está formada por un conjunto de imágenes adquiridas por tres cámaras, las mismas que las utilizadas para la adquisición de la secuencia 1. En estas imágenes aparecen dos robots móviles. En la Figura 9.2 se muestra una escena en la que se pueden observar ambos robots desde las tres cámaras consideradas.



Figura 9.2. Escena perteneciente a la secuencia 2 adquirida por tres cámaras en el ISPACE-UAH.

Tanto para la secuencia 1 como para la 2, se cuenta con las medidas de los sensores odométricos ubicados a bordo de los robots móviles. Indicar que, en las diferentes pruebas realizadas, se ha verificado que las medidas proporcionadas por los sensores odométricos de los robots utilizados para la secuencia 2 presentan un error de medida notablemente menor a las del robot utilizado de la secuencia 1. Ello se debe a que las ruedas del robot utilizado en la secuencia 2 tienen unas características de adherencia que minimizan la posibilidad de patinar sobre el suelo del ISPACE-UAH.

La secuencia 3 es similar a la secuencia 1 ya que está formada por un conjunto de imágenes, adquiridas por cuatro cámaras, en las que puede observarse un robot móvil. La principal diferencia entre ambas es que, en la secuencia 3 existen cambios importantes en la iluminación de la escena. En la Figura 9.3 se pueden observar tres de las cuatro vistas de dos escenas pertenecientes a esta secuencia. En esta figura se aprecian las diferencias en la iluminación entre ambas escenas.

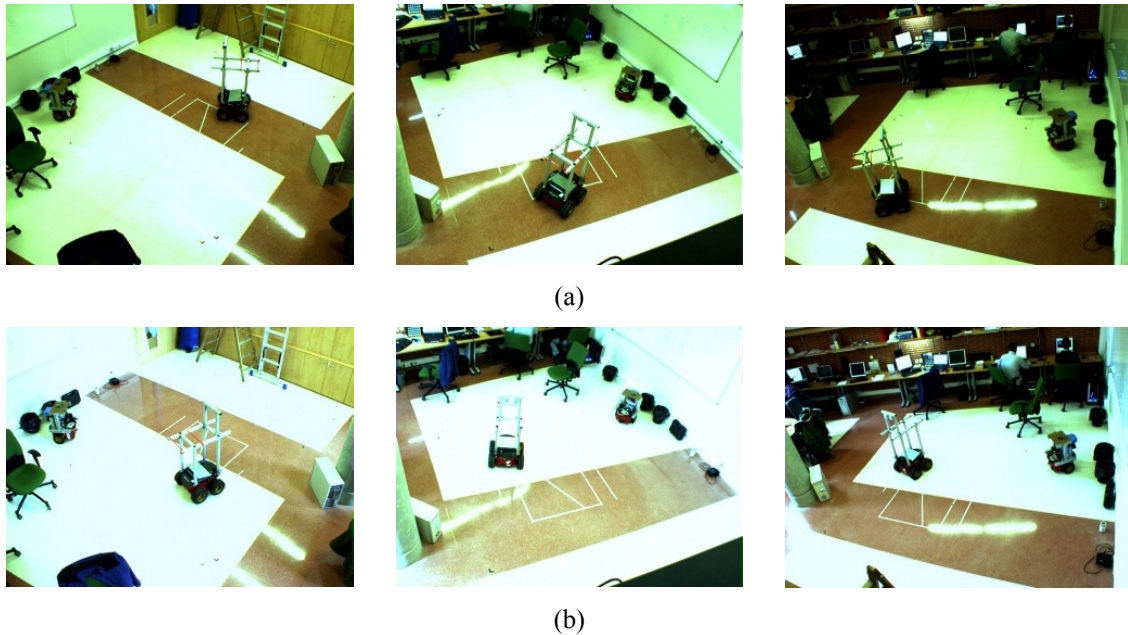


Figura 9.3. Imágenes pertenecientes a la secuencia 3 adquiridas por cuatro cámaras en el ISPACE-UAH. Vista desde tres de las cuatro cámaras de dos escenas en las que aparece un robot móvil. Las dos escenas han sido adquiridas en diferentes condiciones de iluminación.

La secuencia 4 está compuesta por un conjunto de imágenes adquiridas por cuatro cámaras. En esta secuencia se incluyen tanto imágenes en las que sólo aparece un robot móvil, como otras en las que aparece, además, una persona. En la Figura 9.4 se muestran dos escenas pertenecientes a la secuencia 4 vistas desde tres de las cuatro cámaras. En la primera escena (Figura 9.4(a)) se puede apreciar un robot y una persona, mientras que en la segunda (Figura 9.4(b)) únicamente aparece el robot móvil.

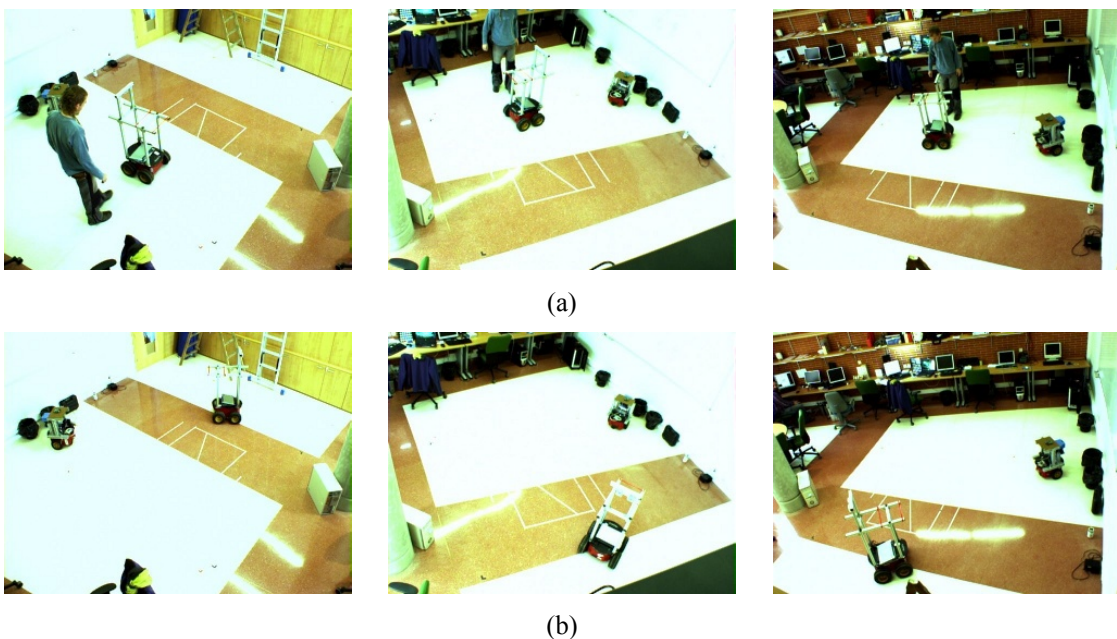


Figura 9.4. Imágenes pertenecientes a la secuencia 4 adquiridas por cuatro cámaras en el ISPACE-UAH. Vista desde tres de las cuatro cámaras de dos escenas diferentes. (a) Imagen en la que aparece un robot móvil y una persona. (b) Imagen en la que únicamente aparece un robot móvil.

Finalmente, las imágenes incluidas en la secuencia 5 han sido adquiridas por 4 cámaras. En esta secuencia se tienen tanto imágenes en las que sólo aparece un robot móvil, como otras en las que aparece además una persona realizando tareas de colocación de obstáculos dentro del área de movimiento del robot. En la Figura 9.5 se muestran tres de las cuatro vistas disponibles de dos escenas pertenecientes a esta secuencia. Cabe destacar que en el caso de la secuencia 5 no se tienen disponibles las medidas de los sensores odométricos a bordo del robot.

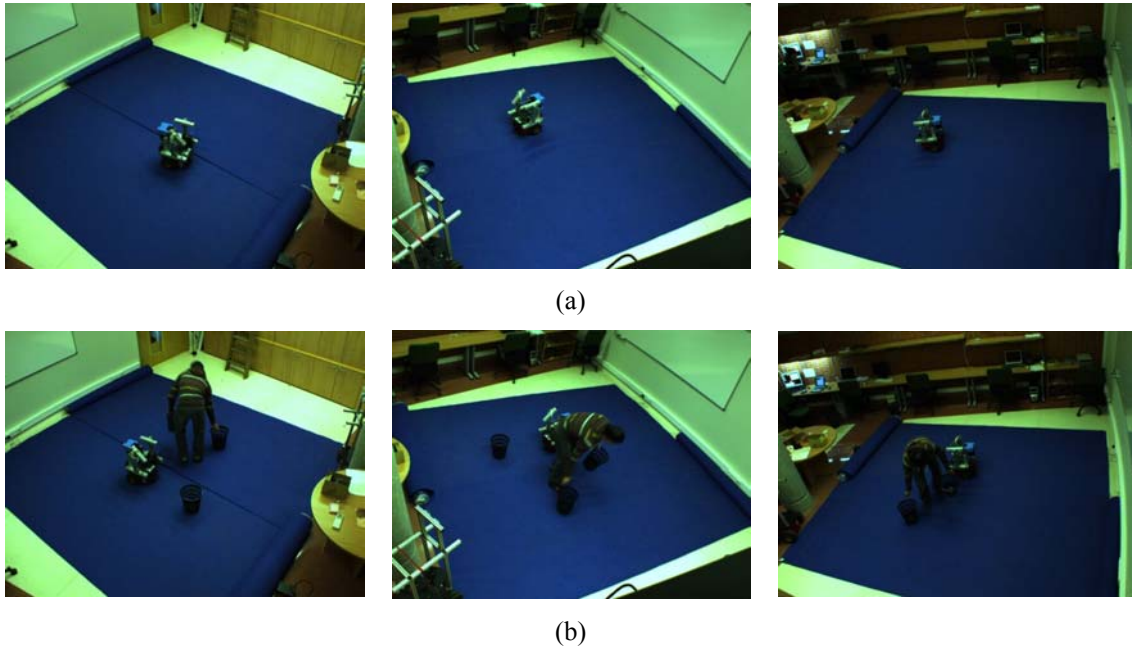


Figura 9.5. Imágenes pertenecientes a la secuencia 5 adquiridas por cuatro cámaras en el ISPACE-UAH. Vista desde tres de las cuatro cámaras de dos escenas diferentes. (a) Imagen en la que únicamente aparece el robot móvil (b) Imagen en la que se muestra tanto un robot como una persona que coloca diferentes obstáculos en el área de movimiento del robot.

Para finalizar, mencionar que las medidas de posición obtenidas a partir de los sensores odométricos de los robots utilizados en las secuencias 1 y 4 presentan un error de medida mayor que los de los robots utilizados para las secuencias 2 y 3, debido a las diferentes características de adherencia de las ruedas (lo que hace que en algunos casos existan deslizamientos notables).

## 9.2. Inicialización de variables

En este apartado se presenta una serie de pruebas que ha permitido validar las diferentes soluciones propuestas en esta tesis para la etapa de inicialización de variables. Los resultados presentados a continuación ponen de manifiesto la importancia de la

correcta inicialización de las curvas que definen los contornos de los robots móviles y la profundidad. Esto se debe a la fuerte dependencia del algoritmo frente a los valores iniciales de estas variables, que determinan tanto el tiempo de cómputo como la calidad de la segmentación y la posición estimada.

Para las pruebas experimentales presentadas en este apartado se han utilizado las secuencias de imágenes 1 (Figura 9.1) y 2 (Figura 9.2) descritas en el apartado 9.1.

### **9.2.1. Inicialización de las curvas**

Para comenzar, se analiza la mejora producida al realizar la inicialización de las curvas mediante GPCA, seguida de un proceso de eliminación de las sombras, tal como se ha expuesto en el capítulo 5 de esta memoria. Se comienza comparando los resultados obtenidos utilizando la técnica propuesta en este trabajo, con los que se consiguen al inicializar las curvas utilizando circunferencias. Además, se analiza también la robustez del método de inicialización elegido frente a cambios en la iluminación de la escena.

Para elegir la ubicación de las circunferencias que definen en principio los contornos de los robots se ha aplicado un umbral sobre la derivada temporal de las imágenes de entrada seguido de un proceso de clasificación basado en  $k$ -medias. El centro de cada circunferencia se ubica en el centroide del grupo de puntos correspondiente, mientras que el radio se elige en función de la dispersión de las muestras que pertenecen a la clase considerada.

Todos los resultados presentados en este apartado han sido obtenidos utilizando el algoritmo de segmentación y posicionamiento 3D para una única cámara, descrito en el capítulo 4, y que está basado en la minimización de la función objetivo definida en [Sekkati & Mitiche 2006b].

En las pruebas experimentales presentadas en este apartado se han utilizado dos secuencias de imágenes diferentes, en concreto las secuencias 1 (Figura 9.1) y 2 (Figura 9.2). De cada una de estas secuencias se han elegido 100 imágenes consecutivas, para las que se muestran los resultados obtenidos. Estas imágenes se procesan en pares ya que para la segmentación es necesario obtener la derivada temporal de las imágenes de entrada.

Para las dos secuencias de imágenes consideradas se han utilizado los mismos valores para las constantes de ponderación de los diferentes términos de la función objetivo ( $\lambda$  y  $\mu$ ). Estos valores han sido fijados de forma experimental para las imágenes de la secuencia 1 tal como se ha expuesto en el capítulo 7 de esta memoria. Cabe destacar que el valor de la constante  $\lambda$  difiere en función del método de inicialización utilizado, ya que determina la velocidad de evolución de las curvas. En concreto, en caso de utilizar circunferencias se ha elegido un valor de  $\lambda$  pequeño ( $\lambda = 10$ ), mientras que para la inicialización usando GPCA se fija un valor de  $\lambda$  grande ( $\lambda = 10.000$ ). Respecto al valor de la constante  $\mu$ , en todos los casos se ha considerado  $\mu=0.003$ . Por otro lado, la profundidad se ha inicializado a un valor constante para todos los puntos de la imagen. El valor elegido ha sido 4m, que es un valor próximo a la distancia real entre el robot y la cámara.

Se comienza con la comparación del tiempo de proceso necesario para obtener la segmentación de movimiento y la posición 3D de cada par de imágenes de entrada de la secuencia 1 (Figura 9.1). Estos valores de tiempo de proceso se muestran en la gráfica de la Figura 9.6(a). Se puede observar que el tiempo consumido en caso de utilizar circunferencias para la inicialización de las curvas es muy superior al caso de emplear la técnica basada en GPCA propuesta en esta tesis. La relación entre los valores de tiempo es del orden de 20. En la Figura 9.6(a) se observa también que, para algunas de las imágenes, el valor de tiempo es muy similar independientemente del tiempo de inicialización utilizada. Esto se debe a que en esas imágenes el robot permanece estático, no siendo detectado en caso de utilizar circunferencias. Esto puede observarse en la Figura 9.6(c) en la que se ha representado el número de robots detectados para cada par de imágenes de entrada entre la 50 y la 70.



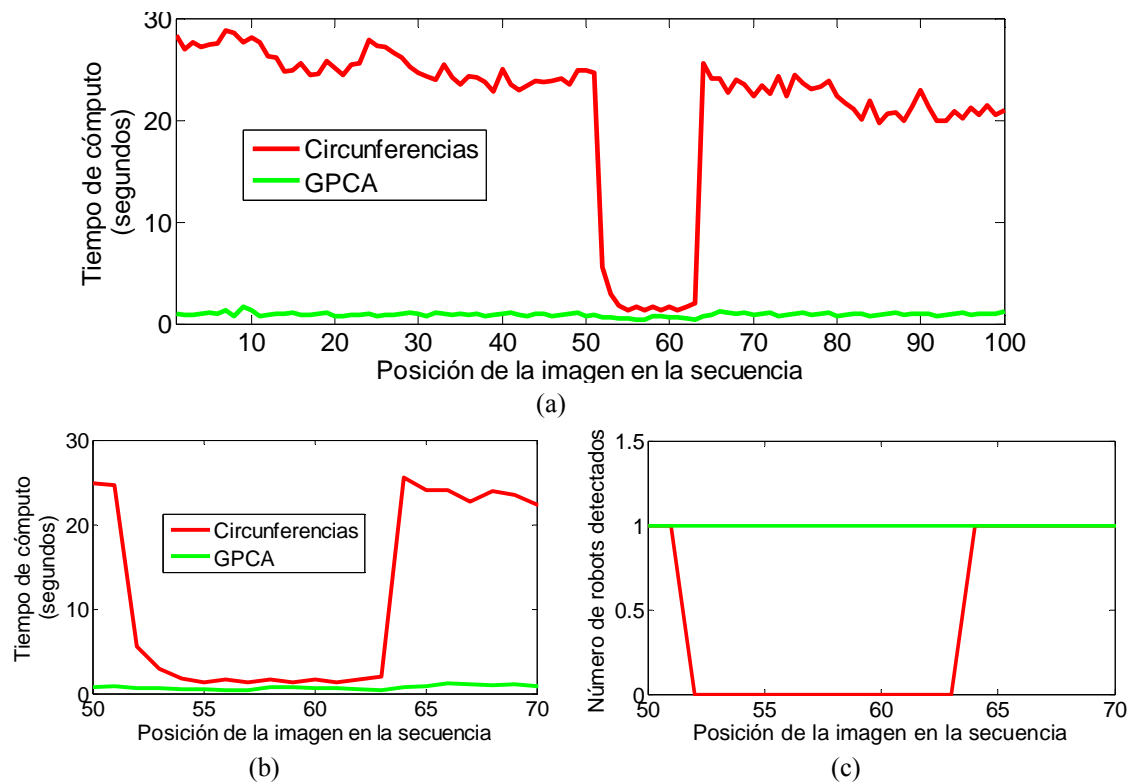


Figura 9.6. (a) Tiempo de cómputo consumido en la inicialización de variables y la minimización de la función objetivo de 100 imágenes de la secuencia 1 (Figura 9.1). El color rojo se corresponde con la inicialización utilizando circunferencias y en verde utilizando la técnica GPCA. (b) Detalle del tiempo de cómputo para las imágenes entre la 150 y la 170. (c) Detalle del número de robots detectados en cada par de imágenes de entrada de la secuencia 1.

En el caso de las imágenes pertenecientes a la secuencia 2 (Figura 9.2), los valores del tiempo consumido por cada par de imágenes se presentan en la Figura 9.7(a). En la Figura 9.7 se puede apreciar que en caso de utilizar inicialización mediante circunferencias, el tiempo consumido es superior al caso de utilizar GPCA. Al igual que en el caso de la secuencia 1, las posiciones en las que los tiempos se aproximan más se deben a que, en caso de inicializar las curvas usando circunferencias, el número de robots detectados ha sido inferior al real.

Cabe destacar que, en caso de inicializar las curvas mediante GPCA, el número de robots detectados es correcto para todas las imágenes, tanto en el caso de la secuencia 1 como en la secuencia 2.

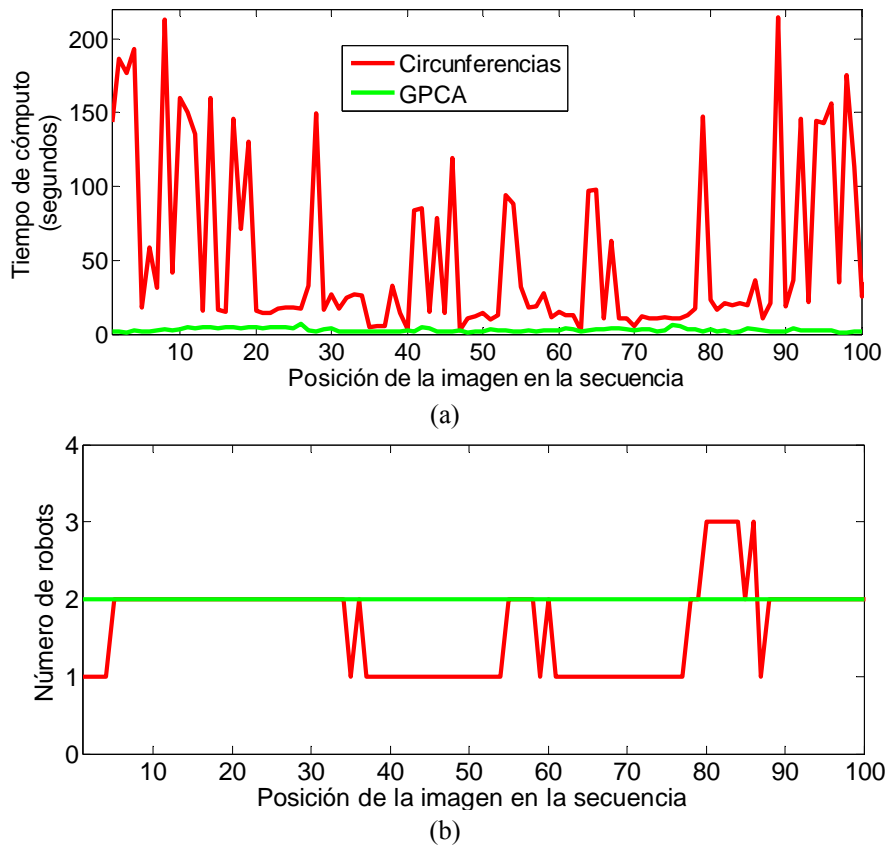


Figura 9.7. (a) Tiempo consumido en la inicialización de variables y la minimización de la función objetivo de 100 imágenes de la secuencia 2. En rojo se muestran los resultados inicializando las curvas de contorno con circunferencias y en verde utilizando GPCA. (b) Número de robots detectados en cada par de imágenes de entrada de la secuencia 1.

A modo de resumen, en la Tabla 9.1 se muestran los valores medios del tiempo de cómputo consumido, y el número de iteraciones hasta la convergencia del algoritmo de minimización. Estos valores han sido obtenidos para 100 imágenes de la secuencia 1 y la secuencia 2.

		Secuencia 1		Secuencia 2	
		Circunf.	GPCA	Circunf.	GPCA
Tiempo de cómputo (segundos)	Inicialización	0.0355	0.1353	0.0404	0.1548
	Minimización	21.3600	0.7711	52.4244	2.7438
	Total	21.3954	0.9064	52.4648	2.8986
n° medio de iteraciones		61.4700	1.9600	79.5481	4.0000

Tabla 9.1. Tiempo de cómputo medio (en segundos) consumido en las etapas de inicialización y minimización de la función objetivo y número medio de iteraciones necesarias para 100 imágenes de las secuencias 1 y 2, en función de las curvas iniciales utilizadas.

En la Tabla 9.1 se aprecia que, aunque el tiempo de cómputo necesario para la inicialización de las curvas mediante GPCA es superior al consumido en caso de utilizar circunferencias, el tiempo total es inferior para GPCA. Esto se debe a que el número de

iteraciones necesarias para la convergencia del algoritmo de minimización de la función objetivo es inferior en caso de inicializar las curvas mediante GPCA, ya que estas curvas están más próximas a los contornos reales de los robots móviles que las circunferencias. Esto se puede observar en las imágenes de la Figura 9.8, en las que se han representado las curvas iniciales (en rojo) y el resultado de la segmentación de movimiento (en verde) para una imagen perteneciente a la secuencia 1. Además, incluso tras un número elevado de iteraciones (59 en el caso de la Figura 9.8(a)), la calidad de la segmentación es superior al inicializar las curvas mediante GPCA.

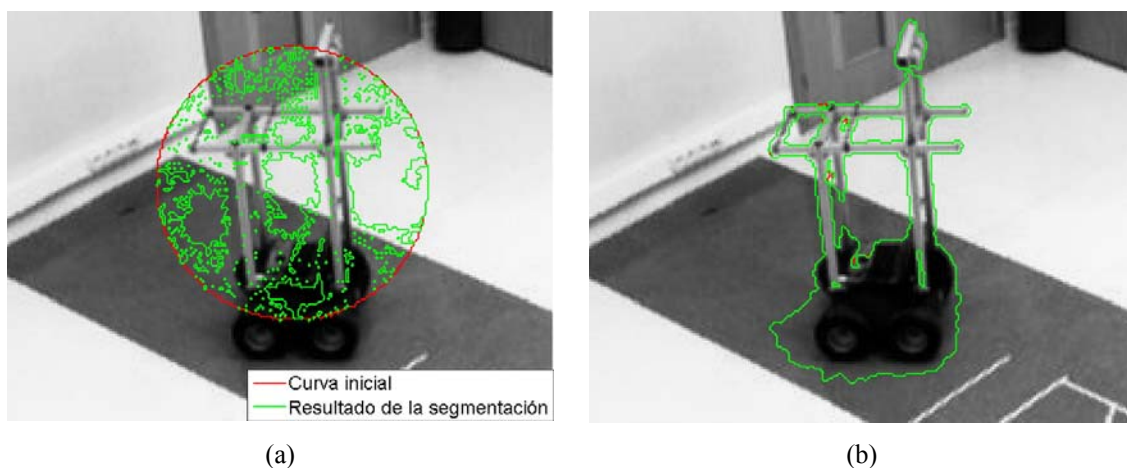


Figura 9.8. Detalle de las curvas iniciales (en rojo) y el resultado de la segmentación de movimiento (en verde) para la imagen 85 de la secuencia 1 (a) Al inicializar las curvas usando circunferencias (resultado obtenido tras 59 iteraciones). (b) Al inicializar las curvas mediante GPCA (resultado obtenido tras 2 iteraciones). Gran parte de la curva inicial no se puede ver porque el resultado de la segmentación se superpone a ella.

Tras evaluar el efecto de las curvas iniciales sobre el tiempo de cómputo y la calidad de la segmentación, a continuación se presentan los experimentos realizados para determinar el efecto de las mismas sobre la robustez del sistema frente a cambios en la iluminación. Para ello se han utilizado 100 imágenes pertenecientes a la secuencia 3 (Figura 9.3), debido a que es la única que ha sido adquirida en condiciones de iluminación variable. En la Figura 9.9 se muestra el resultado de la segmentación para algunas de las imágenes pertenecientes a la secuencia 3. En las diferentes imágenes de esta figura se puede observar que, a pesar de las variaciones en la iluminación de la escena, el resultado de la segmentación es correcto en todos los casos.



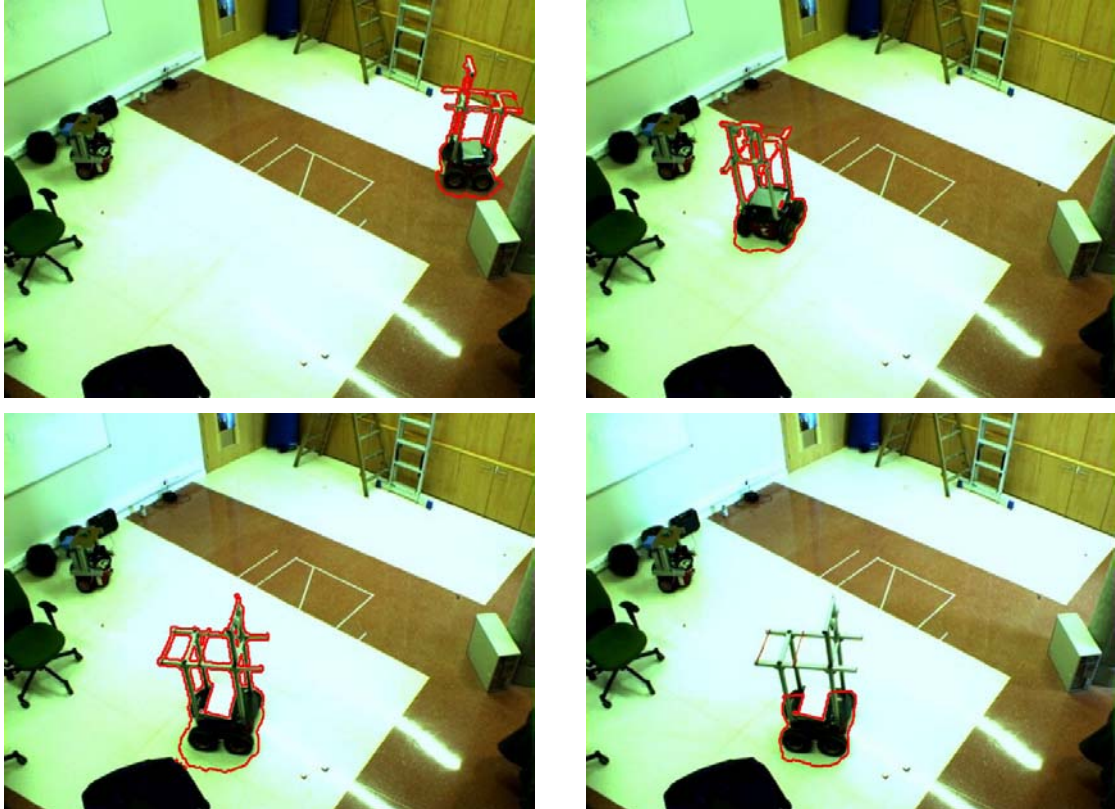


Figura 9.9. Resultado de la segmentación de movimiento de cuatro imágenes pertenecientes a la secuencia 3 (Figura 9.3) que han sido adquiridas en diferentes condiciones de iluminación.

## 9.2.2. Inicialización de la profundidad

El uso de Visual Hull 3D (VH3D) para combinar la información de múltiples cámaras permite obtener una buena estimación de la profundidad inicial. Con el objetivo de validar esta propuesta, se comparan los resultados obtenidos utilizando Visual Hull 3D, con los que se obtienen al inicializar la profundidad a un valor fijo para todos los puntos de la imagen. Para ello se han utilizado de nuevo, las secuencias de imágenes 1 (Figura 9.1) y 2 (Figura 9.2). Para cada una de ellas se evalúa tanto el tiempo de cómputo, como el error cometido en el posicionamiento 3D de los robots móviles. En todos los casos, los resultados presentados se han obtenido utilizando la solución propuesta para la segmentación y posicionamiento de robots móviles a partir de las imágenes adquiridas por tres cámaras en el ISPACE-UAH. En las pruebas realizadas en este apartado no se incluye el XPCFP para el seguimiento de los robots móviles.

En la Figura 9.10 se muestra el error cuadrático cometido en la estimación de la posición 3D de los robots móviles en cada una de las 100 imágenes de la secuencia 1

(Figura 9.1). El valor del error mostrado en las gráficas de la Figura 9.10 se obtiene como la diferencia entre la posición obtenida a partir de los sensores odométricos de los robots y la estimada a partir de las cámaras utilizando el algoritmo propuesto. Esta diferencia se calcula mediante la ecuación (9.1) en la que  $(x_o, y_o)$  representa la posición obtenida a partir de los sensores odométricos y  $(x_e, y_e)$  la estimada por el algoritmo propuesto.

$$\varepsilon_p = \sqrt{(x_o - x_e)^2 + (y_o - y_e)^2} = \sqrt{\varepsilon_{px}^2 + \varepsilon_{py}^2} \quad (9.1)$$

En la Figura 9.10 se puede observar que el valor medio del error de posicionamiento es menor en el caso de utilizar VH3D para la inicialización de la profundidad. Además, los resultados se obtienen en un tiempo menor, tal como se puede observar en la gráfica inferior de la Figura 9.10.

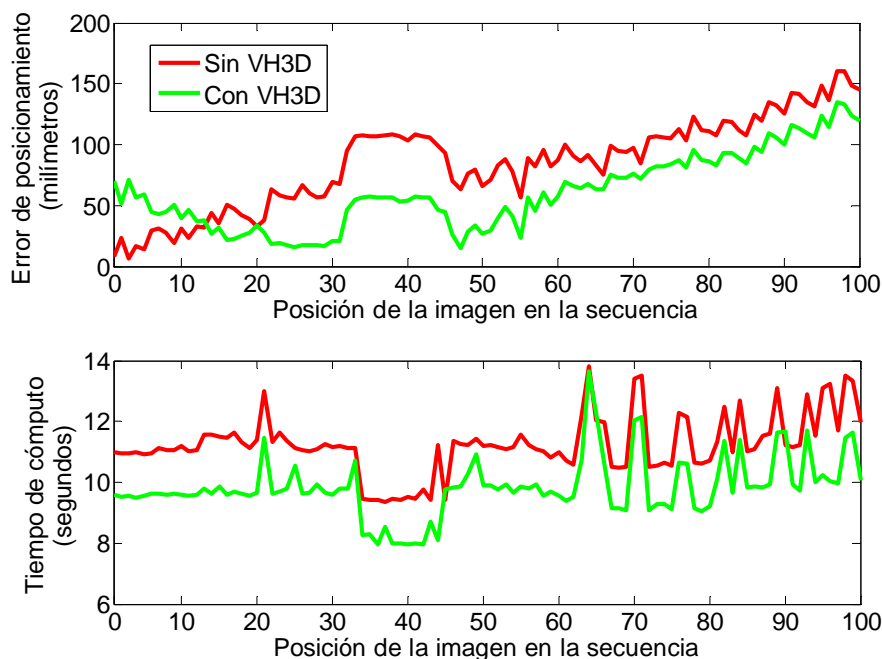


Figura 9.10. Error cuadrático cometido en el posicionamiento de los robots móviles (parte superior) y tiempo de procesamiento (parte inferior) de cada par de imágenes de la secuencia 1.

Para el caso de las imágenes de la secuencia 2, en la Figura 9.11 se muestran los resultados obtenidos para 100 imágenes de la misma. Debido a que en la segunda secuencia se tienen dos robots móviles, el error de posicionamiento del primero de ellos se ha representado con línea continua, mientras que para el segundo se utiliza línea discontinua. Se puede observar como, en caso de inicializar la profundidad mediante Visual Hull 3D, el valor medio del error de posicionamiento es inferior, siendo también

menor el tiempo de procesamiento necesario hasta la convergencia del algoritmo, tal como se puede observar en la Figura 9.11(b).

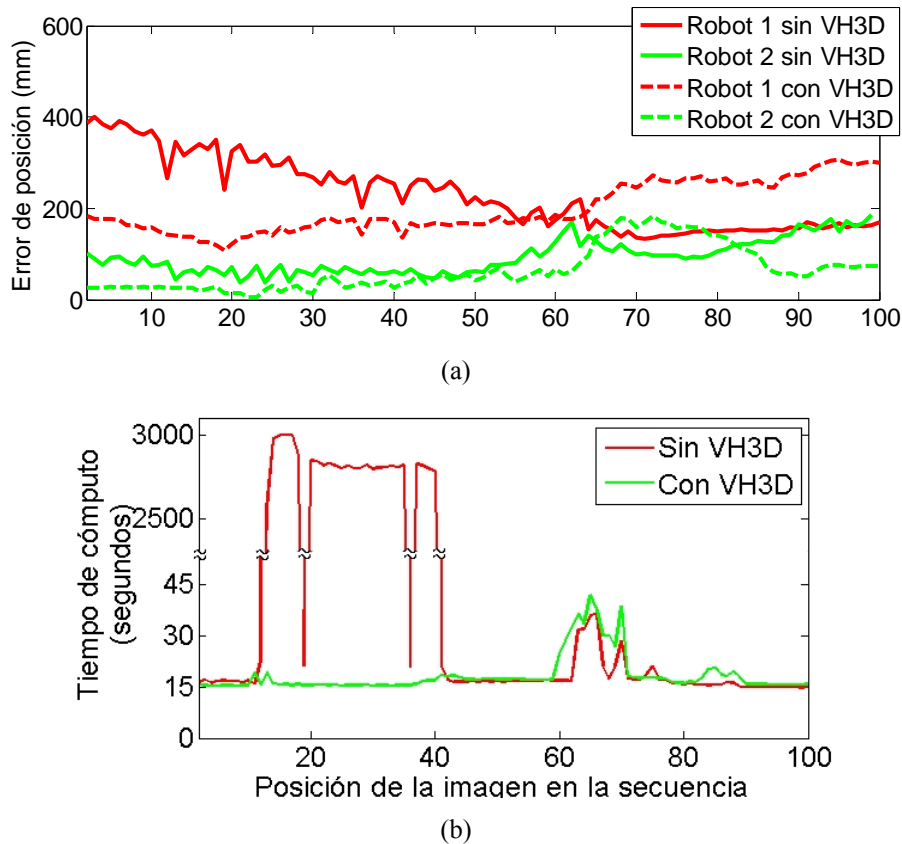


Figura 9.11. (a) Error cuadrático cometido en el posicionamiento de los robots móviles (b) Tiempo de procesamiento de cada par de imágenes de la secuencia 2.

A continuación, en la Figura 9.12 se muestra la trayectoria estimada para cada uno de los robots móviles de la secuencia 2, en función del método de inicialización de la profundidad considerado. En color verde se muestra la trayectoria estimada en caso de utilizar VH3D en la inicialización de la profundidad, y en rojo para el caso de suponer que todos los puntos pertenecientes al robot se encuentran a la misma distancia. En esta figura se aprecia fácilmente que las posiciones estimadas al usar VH3D en la inicialización están más próximas a las medidas obtenidas a partir de los sensores odométricos a bordo de los robots móviles.

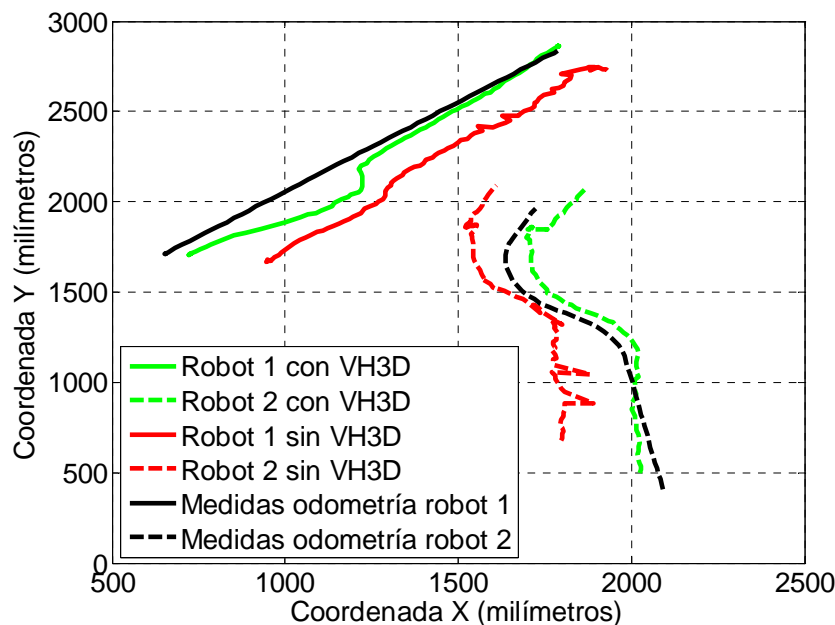


Figura 9.12. Trayectoria de los robots móviles a lo largo de las imágenes de la secuencia 2. Resultado de la proyección sobre el plano XY en  $\Gamma_w$ .

En la Tabla 9.2 se presentan, a modo de resumen, los valores medios del error de posicionamiento (calculado mediante la ecuación (9.1)) y el tiempo de procesamiento de las 100 imágenes de las secuencias 1 y 2. En el caso de la secuencia 2, se muestran los valores medios del error para ambos robots.

		Secuencia 1		Secuencia 2	
		Planos	VH3D	Planos	VH3D
Tiempo de cómputo (segundos)	Inicialización	3.8634	3.1767	4.9764	4.5844
	Minimización	7.3941	6.6721	766.2485	13.8273
	Total	11.2575	9.8488	771.2249	18.4118
Error de posición (mm)		76.2633	66.7977	231.59 199.52	92.84 76.14

Tabla 9.2. Error de posicionamiento medio (en milímetros) y tiempo de cómputo medio (en segundos) consumido en las etapas de inicialización y minimización de la función objetivo obtenidas para 100 imágenes de las secuencias 1 y 2, en función del método de inicialización de la profundidad utilizado.

### 9.3. Segmentación de movimiento y estimación de la posición 3D de los robots móviles

A lo largo del capítulo 4 se han presentado dos funciones objetivo (F.O.) tales que a partir de su minimización es posible obtener la segmentación de movimiento y estimar la posición 3D de múltiples robots móviles. La primera de estas funciones es la

propuesta en el trabajo de [Sekkti & Mitiche 2006b], que incorpora la información de una única cámara mientras que, la segunda incorpora la información de varias cámaras y es una de las aportaciones significativas de esta tesis. En este apartado se presentan los resultados obtenidos con ambas funciones. Esto se hace con el objetivo de validar las dos soluciones planteadas en la tesis para la segmentación de movimiento y el posicionamiento 3D de los robots móviles.

En las pruebas experimentales presentadas a continuación se han utilizado 100 pares de imágenes pertenecientes a las secuencias 1 (Figura 9.1) y 2 (Figura 9.2). De nuevo, debido a que es necesario calcular la derivada temporal de las imágenes de entrada, estas se procesan de dos en dos.

Para cada una de las secuencias se ha ejecutado la solución propuesta para la segmentación y posicionamiento de robots móviles a partir de las imágenes adquiridas por 1, 2 y 3 cámaras, sin utilizar el XPFCP para el seguimiento de los robots. En los casos en que se utiliza la función para una única cámara, los resultados finales se obtienen combinando los obtenidos para cada una de las cámaras por separado, tal como se ha expuesto en el capítulo 4 de esta memoria.

Cabe destacar que, con el objetivo de poder comparar los resultados obtenidos utilizando ambas funciones, tanto los valores iniciales de las variables implicadas (curvas de contorno y profundidad) como el criterio de convergencia del proceso de minimización son iguales independientemente de la función objetivo considerada en cada caso.

En todos los experimentos realizados se ha evaluado tanto el tiempo de cómputo, como la precisión en el posicionamiento 3D de los robots móviles. El error cometido en el posicionamiento en cada imagen se ha calculado utilizando la expresión mostrada en la ecuación (9.1).

Se comienza presentando los resultados (tiempo de cómputo y error de posición) obtenidos en caso de utilizar las imágenes adquiridas por una única cámara. En la Figura 9.13 se muestra tanto el tiempo de cómputo, como el error cometido en la estimación de la posición 3D usando ambas funciones objetivo para 100 pares de imágenes pertenecientes a la secuencia 1 (Figura 9.1). En esta figura se puede observar que, en caso de utilizar una única cámara, el error de posicionamiento toma valores muy próximos para ambos tipos de funciones objetivo, ya que en este caso ambas son muy

similares (la única diferencia está en los sistemas de referencia en los que se representan las velocidades). Respecto al tiempo de cómputo, éste es superior en caso de utilizar la función objetivo para una cámara. En la gráfica inferior de la Figura 9.13 también se puede observar que, en caso de utilizar una única cámara, el error de posición es muy elevado (con un valor medio de 1500 milímetros). Esto es debido a que, al utilizar una única cámara no es posible recuperar la información de profundidad. Además, en este caso no puede utilizarse VH3D para la inicialización de la profundidad, de forma que ésta se inicializa a un valor constante para todos los puntos pertenecientes al robot.

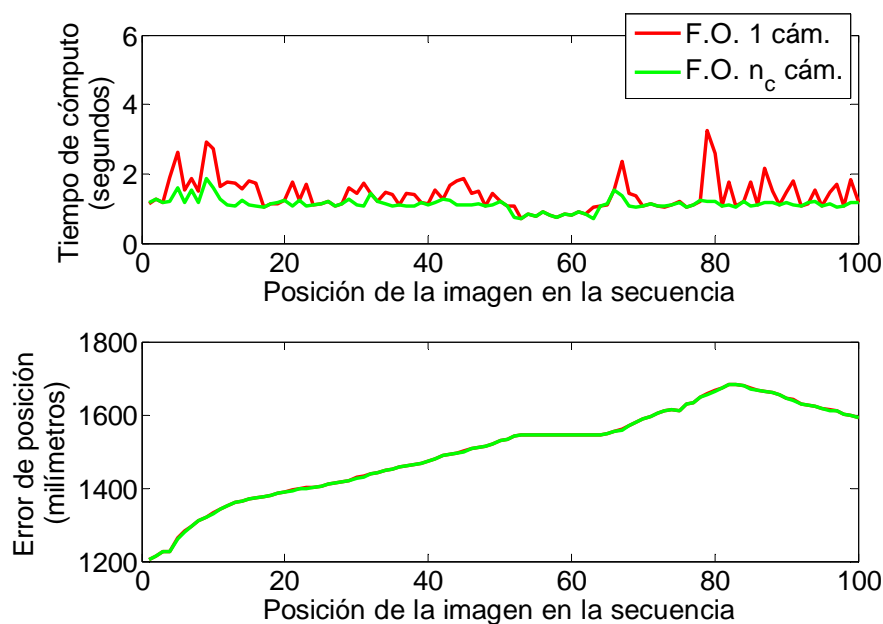


Figura 9.13. Tiempo de procesamiento (gráfica superior) y error de posición (gráfica inferior) para 100 imágenes pertenecientes a la secuencia 1 (Figura 9.1). En rojo se muestran los resultados para el caso de la F.O. para una sola cámara y en verde para la F.O. para múltiples cámaras. En el caso del error de posición, las gráficas para ambas F.O. coinciden. .

En la Figura 9.13 se muestra el tiempo de cómputo y el error cometido en el posicionamiento (calculado mediante la ecuación (9.1)) para ambas funciones objetivo, en caso de utilizar 100 pares de imágenes pertenecientes a la secuencia 2 (Figura 9.2) adquiridas por una única cámara. Al igual que en el caso de la secuencia 1, en esta figura se aprecia que los resultados obtenidos son muy parecidos, independientemente de la función objetivo utilizada. Además, debido al uso de una única cámara, el error de posicionamiento es bastante elevado.

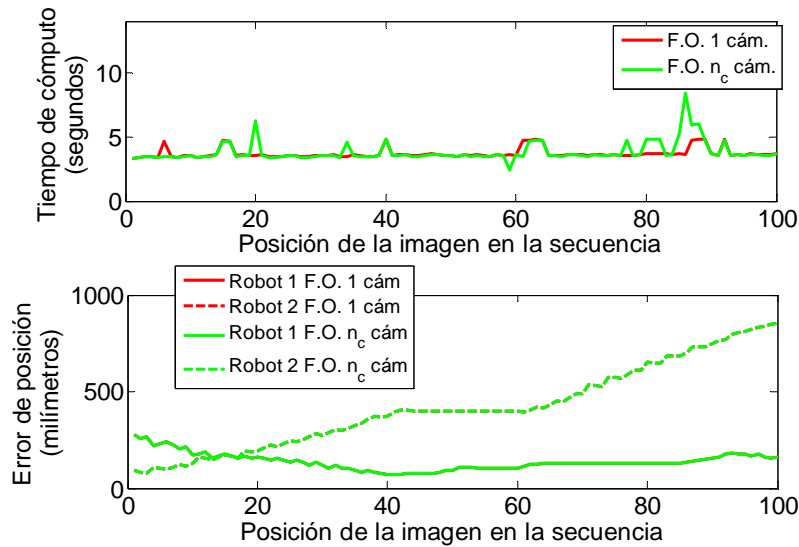


Figura 9.14. Tiempo de procesamiento (gráfica superior) y error de posición (gráfica inferior) para 100 imágenes pertenecientes a la secuencia 2 (Figura 9.2) obtenidas a partir de las imágenes adquiridas por una única cámara. En el caso del error de posición, las gráficas obtenidas para ambas funciones objetivo se superponen.

En la Figura 9.15 se muestran los resultados obtenidos utilizando las imágenes pertenecientes a la secuencia 1 (Figura 9.1) y adquiridas por 2 (columna izquierda) y tres cámaras (columna derecha). Al igual que en el caso de una única cámara, el error de posicionamiento es muy similar para ambas funciones objetivo, mientras que el tiempo de cómputo es superior para el caso de la función objetivo de [Sekkti & Mitiche 2006b].

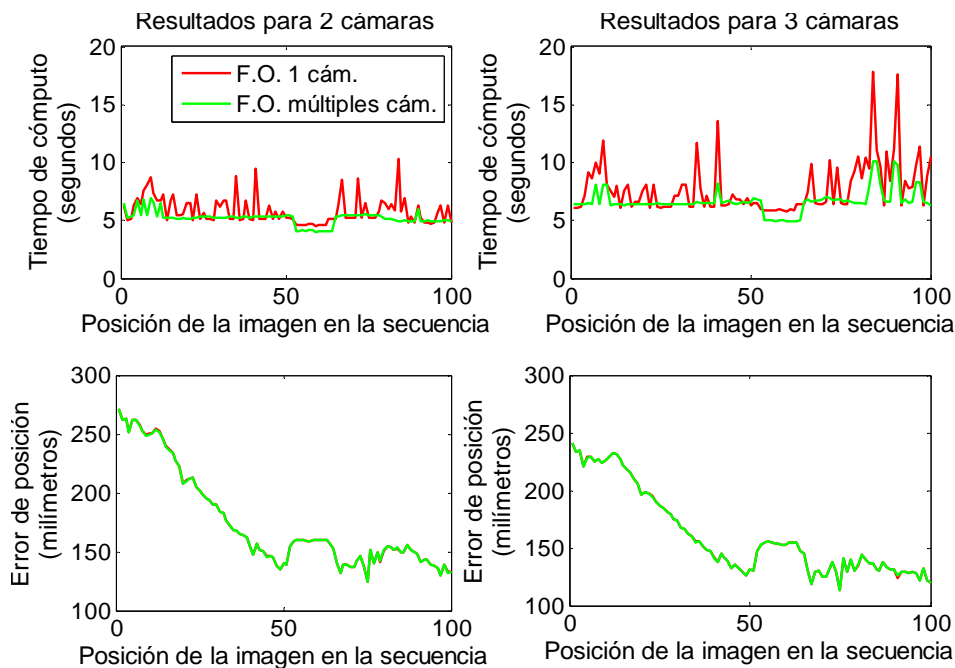


Figura 9.15. Tiempo de procesamiento (gráfica superior) y error de posición (gráfica inferior) para 100 imágenes pertenecientes a la secuencia 1 (Figura 9.1) obtenidas a partir de las imágenes adquiridas por dos (columna izquierda) y tres (columna derecha) cámaras. En el caso del error de posición, las gráficas para ambas funciones objetivo coinciden.

La gran similitud de los valores obtenidos en la estimación de la posición 3D (mostrada en la Figura 9.15) se debe al uso de VH3D para la inicialización de la profundidad en ambos casos. Mediante VH3D se combina la información de las diferentes cámaras. De esta forma, a pesar de que la función objetivo definida para una cámara no incorpora información de las demás cámaras disponibles, la profundidad inicial sí incluye dicha información, mejorando notablemente los resultados obtenidos.

En el caso de la secuencia 2, las conclusiones extraídas son similares. En la Figura 9.16 se muestran los resultados obtenidos para 100 pares de imágenes de la secuencia 2 (Figura 9.2) adquiridas por dos y tres cámaras.

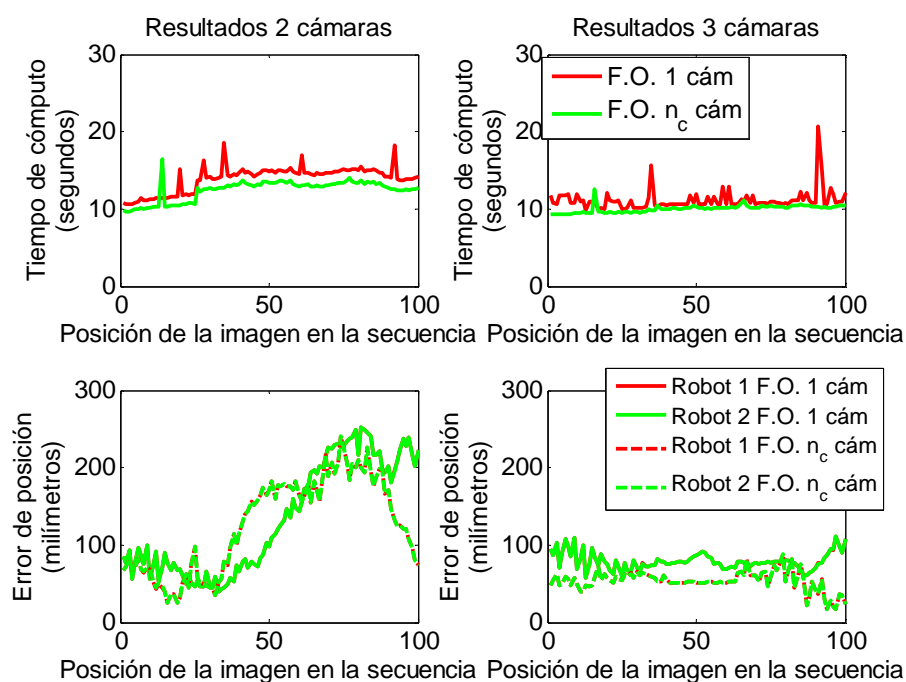


Figura 9.16. Tiempo de procesamiento (gráficas superiores) y error de posición (gráficas inferiores) para 100 imágenes pertenecientes a la secuencia 2 (Figura 9.2) obtenidos a partir de las imágenes adquiridas por dos (columna izquierda) y tres (columna derecha) cámaras.

Al igual que en el caso de una única cámara, el error de posicionamiento es muy similar para ambas funciones, mientras que el tiempo de cómputo es superior para el caso de la función objetivo de [Sekkati & Mitiche 2006b]. En estas figuras también se puede observar como, a medida que se incrementa el número de cámaras, desciende el valor del error de posicionamiento, llegando a ser del orden de 10 centímetros en caso de utilizar tres cámaras.

Observando las figuras anteriores se puede apreciar que el número de cámaras también influye de forma notable tanto en el tiempo de cómputo, como en la precisión de los resultados. Esto se observa también en la Tabla 9.3 y la Tabla 9.4 en las que, a



modo de resumen, se presentan los valores medios del tiempo de procesamiento de las etapas de la solución propuesta, así como el valor medio del error de posicionamiento obtenido para 100 pares de imágenes de la secuencia 1 y la secuencia 2 respectivamente. En ambas tablas, por cada número de cámaras, la columna izquierda corresponde a los resultados obtenidos con la función objetivo para una cámara (f.o. 1c) mientras que la columna derecha muestra los resultados de la función objetivo para múltiples cámaras (f.o. mc).

		1 cámara		2 cámaras		3 cámaras	
		f.o. 1c	f.o. mc	f.o. 1c	f.o. mc	f.o. 1c	f.o. mc
Tiempo de cómputo (s)	Inicialización	0.4600	0.4600	2.7239	2.7239	3.0303	3.0303
	Segmentación	0.9324	0.6614	2.9585	2.0561	4.2531	2.8661
	Fusión	-	-	0.1892	-	0.2200	-
	Total	1.3924	1.1214	5.8716	4.7800	7.5034	5.8964
Error posición (mm)		1501.60	1501.35	174.77	174.70	160.77	160.76

Tabla 9.3. Valor medio del tiempo de procesamiento y del error cometido en la estimación de la posición 3D, obtenido para 100 imágenes pertenecientes a la secuencia 1, en función del número de cámaras y la función objetivo utilizada.

		1 cámara		2 cámaras		3 cámaras	
		f.o. 1c	f.o. mc	f.o. 1c	f.o. mc	f.o. 1c	f.o. mc
Tiempo de cómputo (s)	Inicialización	1.1348	1.1348	4.6300	4.6300	3.7712	3.7712
	Segmentación	2.5645	2.6746	8.9761	7.8043	7.2138	6.2614
	Fusión	-	-	0.3055	-	0.1525	-
	Total	3.6993	3.8094	13.9115	12.4343	11.1375	10.0326
Error posición (mm)		137.68	137.69	133.99	133.97	79.21	79.22
		418.74	418.74	126.35	126.41	53.53	53.53

Tabla 9.4. Valor medio del tiempo de procesamiento y del error cometido en la estimación de la posición 3D, obtenido para 100 imágenes pertenecientes a la secuencia 2, en función del número de cámaras y la función objetivo utilizada.

En ambas tablas se aprecia que, a medida que el número de cámaras aumenta, el error de posicionamiento disminuye de forma notable. En cuanto al tiempo de procesamiento, este se incrementa notablemente al pasar de 1 a dos cámaras, mientras que se mantiene en valores similares (llegando incluso a disminuir) al pasar de dos a tres cámaras.

Cabe destacar aquí que los resultados obtenidos al utilizar la función objetivo definida en [Sekkati & Mitiche 2006b] dependen fuertemente de los valores iniciales de las curvas y la profundidad. Por este motivo, los resultados obtenidos aplicando ambas

funciones son muy similares. También se han realizado algunas pruebas experimentales inicializando las curvas y la profundidad de forma independiente por cada una de las cámaras, en caso de utilizar la función objetivo para una cámara. Bajo estas condiciones, tanto el tiempo de procesamiento hasta la convergencia, como el error cometido en el posicionamiento 3D se deterioran de forma notable, respecto al caso de la función objetivo para múltiples cámaras.

## 9.4. Seguimiento de múltiples robots móviles

En esta tesis, la etapa de seguimiento se ha implementado mediante un filtro de partículas extendido con proceso de clasificación (XPFCP) [Marron 2008] tal como se ha expuesto en el capítulo 8. El carácter multimodal del XPFCP permite realizar el seguimiento de múltiples robots móviles con un único estimador, sin necesidad de aumentar el vector de estado.

La información obtenida a la salida del XPFCP se realimenta a la etapa de inicialización de variables, ya que supone una buena estimación de la posición que ocuparán los robots móviles en la siguiente imagen de la secuencia. Esta realimentación permite reducir el tiempo consumido por el algoritmo, especialmente durante la etapa de inicialización de variables.

Para comenzar se presentan los resultados obtenidos para 100 imágenes pertenecientes a las secuencias 1 (Figura 9.1) y 2 (Figura 9.2). Estos resultados se obtienen ejecutando el algoritmo de segmentación a partir de las imágenes adquiridas por tres cámaras. Se comparan los valores del tiempo de procesamiento y el error de posicionamiento 3D obtenidos sin XPFCP y con XPFCP con 5.000 partículas.

En la Figura 9.17 se muestra el tiempo de cómputo y el error de posicionamiento para 100 imágenes de la secuencia 1. En esta figura se puede observar como para esta secuencia de imágenes en la que aparece un único robot los resultados son muy similares.

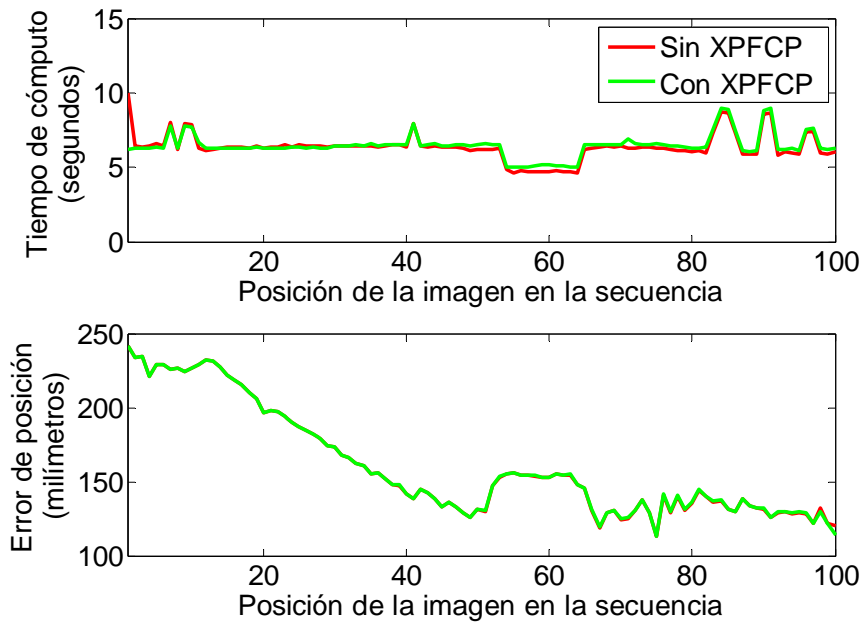


Figura 9.17. Tiempo de cómputo (gráfica superior) y error de posición (gráfica inferior) calculados para 100 imágenes pertenecientes a la secuencia 1 (Figura 9.1).

En el caso de la secuencia 2, en la Figura 9.18 se puede observar como el tiempo de procesamiento es notablemente inferior en caso de incluir el XPFCP para el seguimiento de los robots móviles. Además, el error de posicionamiento también es ligeramente inferior, especialmente en el caso del Robot 2.

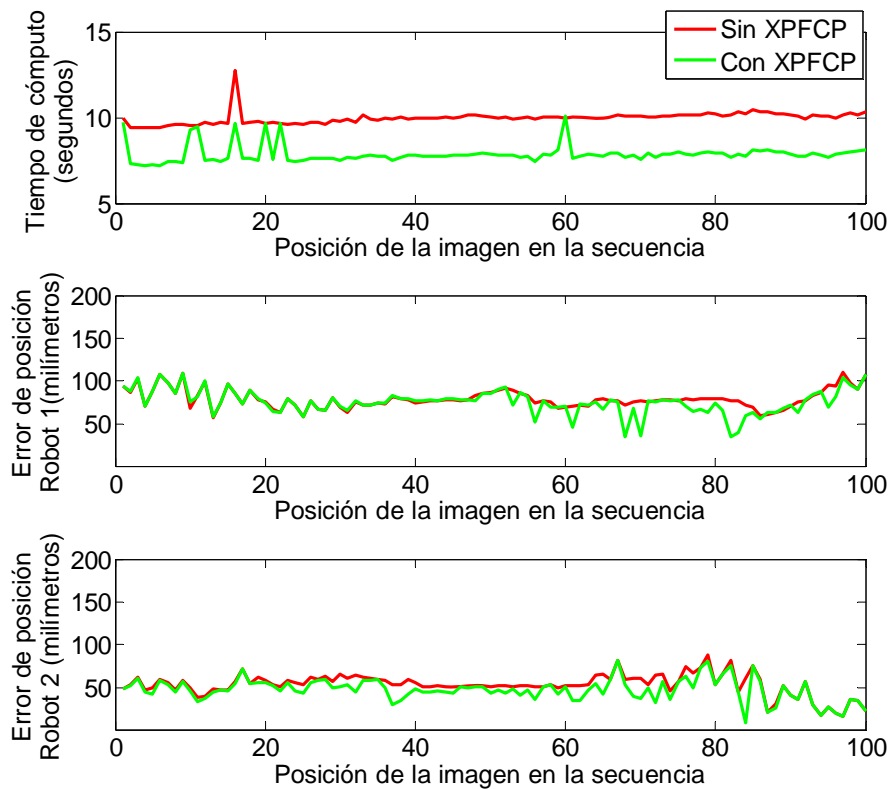


Figura 9.18. Tiempo de cómputo (gráfica superior) y error de posición (gráficas intermedia e inferior) calculados para 100 imágenes pertenecientes a la secuencia 2 (Figura 9.2).

A modo de resumen, en la Tabla 9.5 se presenta el valor medio del tiempo de procesamiento y del error de posicionamiento de cada uno de los casos mostrados en las figuras anteriores.

		Secuencia 1 (Figura 9.1)		Secuencia 2 (Figura 9.2)	
		Sin XPFCP	Con XPFCP	Sin XPFCP	Con XPFCP
Tiempo de cómputo (segundos)	Inicialización	3.4340	3.2387	3.7676	3.3338
	Minimización	2.9609	3.1101	6.2059	4.4823
	Filtrado	-	0.0499	-	0.0576
	Total	6.4448	6.3488	9.9735	7.8737
Error de posición (mm)		160.76	160.87	79.22 53.53	75.60 47.54

Tabla 9.5. Valor medio del tiempo de procesamiento y del error de posicionamiento 3D calculado para 100 imágenes pertenecientes a las secuencias 1 y 2.

#### 9.4.1. Efecto del número de partículas

En el XPFCP, el número de partículas es un parámetro importante que influye notablemente en el tiempo consumido por el filtro. Este número se fija en la configuración del filtro, y permanece constante independientemente del número de robots. Esta es una de las ventajas del XPFCP, ya que permite que el tiempo de procesamiento permanezca constante a pesar de que el número de robots sea variable.

Con el objetivo de determinar el efecto del número de partículas sobre el tiempo de procesamiento y la precisión de los resultados del algoritmo de segmentación y posicionamiento 3D propuesto en la tesis, se han realizado diferentes pruebas experimentales. Los resultados de las mismas se describen brevemente a continuación.

Se comienza analizando el efecto del número de partículas en el tiempo consumido por el XPFCP. Para ello, se ha ejecutado la solución propuesta para la segmentación de movimiento y posicionamiento 3D sobre 100 imágenes pertenecientes a las secuencia 1 (Figura 9.1) y 2 (Figura 9.2), y se ha obtenido el valor medio del tiempo para diferentes valores del número de partículas. Estos tiempos medios se muestran gráficamente en la Figura 9.19. En esta figura se puede observar como, a medida que aumenta el número de partículas, se incrementa el tiempo consumido por el XPFCP. Además, también se aprecia que el valor del tiempo medio es muy similar para las dos secuencias de

imágenes consideradas, ya que presenta una dependencia mayor frente al número de partículas que frente a los datos de entrada o el número de robots móviles.

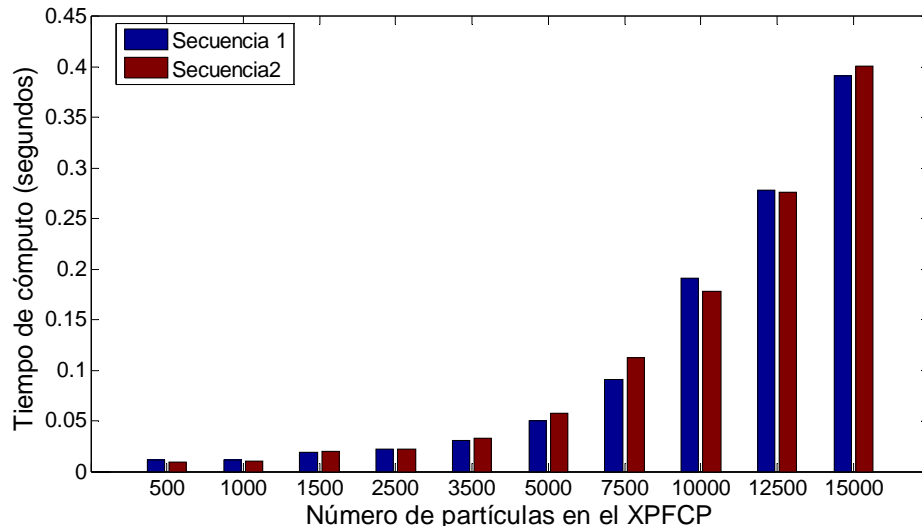


Figura 9.19. Valor medio del tiempo consumido por el XPFCP, obtenido para 100 imágenes pertenecientes a las secuencia 1 (Figura 9.1) y 2 (Figura 9.2).

Sin embargo, el incremento en el tiempo consumido por el XPFCP no se traduce en un aumento del tiempo total del algoritmo, tal como se puede observar en la Figura 9.20. En esta figura se ha representado el tiempo medio del proceso de segmentación y posicionamiento 3D de los robots móviles para 100 imágenes de cada una de las secuencias de test.

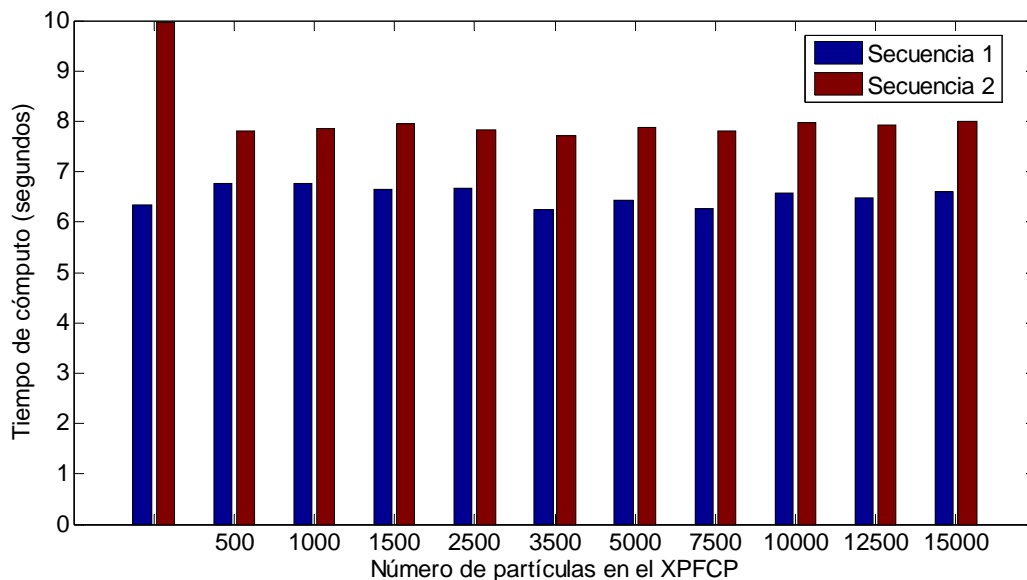


Figura 9.20. Tiempo medio consumido para la segmentación de movimiento y la estimación de la posición 3D para 100 imágenes pertenecientes a las secuencias 1 y 2 en función del número de partículas consideradas en el XPFCP.

En la Figura 9.20.se puede observar que este tiempo no es proporcional al número de partículas sino que se mantiene en valores similares, independientemente del número

de partículas. Esto se debe, a que a medida que se incrementa el número de partículas, los resultados obtenidos a la salida del XPFCP son más próximos a las posiciones reales de los robots, consiguiendo disminuir el tiempo consumido en el resto de etapas.

Respecto al error cometido en el posicionamiento 3D, en la Figura 9.21 se muestran los errores medios calculados como la diferencia entre la posición estimada y la medida por los sensores odométricos a bordo de los robots móviles. En esta figura se aprecia como en el caso de la secuencia 1 (en la que aparece un único robot móvil), el valor medio del error de posicionamiento es similar en todos los casos, esto se debe a que al existir un único robot, se obtienen resultados adecuados a pesar de que el número de partículas sea pequeño, de forma que el incremento del número de partículas no mejora la precisión del posicionamiento. Sin embargo, en la gráfica correspondiente a la secuencia 2 (gráfica inferior), el valor del error de posicionamiento disminuye ligeramente a medida que se incrementa el número de partículas del XPFCP.

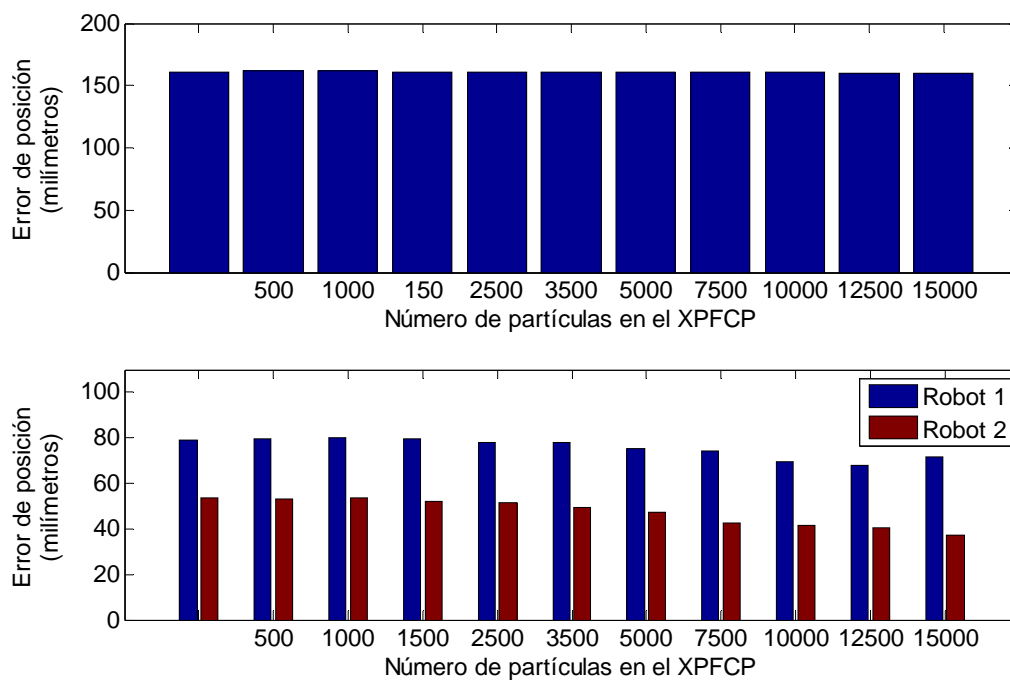


Figura 9.21. Error de posición medio calculado para 100 imágenes pertenecientes a la secuencia 1 (gráfica superior) y a la secuencia 2 (gráfica inferior) en función del número de partículas consideradas en el XPFCP.

A modo de resumen, en la Tabla 9.6 se muestra el valor medio del tiempo de procesamiento y del error de posición en función del número de partículas del XPFCP. Estos valores han sido calculados para 100 imágenes pertenecientes a las secuencias de test 1 y 2.

Número de partículas del XPFCP	Tiempo de cómputo (segundos)		Error de posición (milímetros)		
	Secuencia 1	Secuencia 2	Secuencia 1	Secuencia 2	
<b>Sin XPFCP</b>	6.3488	9.9735	160.7647	79.2242	53.5293
<b>500</b>	6.7672	7.8181	162.0974	79.8220	53.344
<b>1000</b>	6.7572	7.8632	161.3834	79.9345	53.4476
<b>1500</b>	6.6403	7.9503	161.1965	79.4101	51.8755
<b>2500</b>	6.6699	7.8408	161.1405	78.1094	51.3472
<b>3500</b>	6.2383	7.7072	160.9418	77.9214	49.5136
<b>5000</b>	6.4448	7.8737	160.8701	75.5998	47.5445
<b>7500</b>	6.2685	7.8099	160.6617	74.5608	42.5665
<b>10000</b>	6.5852	7.9708	160.5742	69.4625	41.5307
<b>12500</b>	6.4781	7.9263	160.2044	68.0953	40.6485
<b>15000</b>	6.5982	7.9985	159.8362	71.6336	37.0996

Tabla 9.6. Valor medio del tiempo de procesamiento y del error de posición en función del número de partículas del XPFCP, calculados para 100 imágenes pertenecientes a las secuencias 1 y 2.

## 9.5. Validación del sistema completo

Para finalizar, se presentan los resultados obtenidos al utilizar la solución propuesta para la segmentación de movimiento y la estimación de la posición 3D de múltiples robots móviles. De esta forma se pretende validar las diferentes propuestas realizadas a lo largo de la tesis de forma global.

Para las pruebas presentadas en este apartado se han utilizado imágenes pertenecientes a las cinco secuencias descritas en el apartado 9.1.

Para comenzar se presentan los resultados obtenidos en la segmentación de movimiento. Por cada una de las secuencias se muestran algunas imágenes sobre las que se han representado, en diferentes colores, las curvas que definen los contornos de los diferentes objetos detectados. En estas figuras se muestra también la reconstrucción 3D obtenida, usando VH3D, a partir de los contornos de la segmentación, así como la

trayectoria estimada para las diferentes imágenes que forman la secuencia. Dado que la identificación de los robots se realiza después de la segmentación, en esta etapa se detectan todos los elementos móviles presentes en la imagen (tanto robots, como usuarios y obstáculos).

En la Figura 9.22 se muestran los resultados obtenidos para dos imágenes pertenecientes a la secuencia 1 (Figura 9.1). En esta figura se puede observar como tras la segmentación de movimiento se obtiene de forma precisa el contorno del robot móvil.

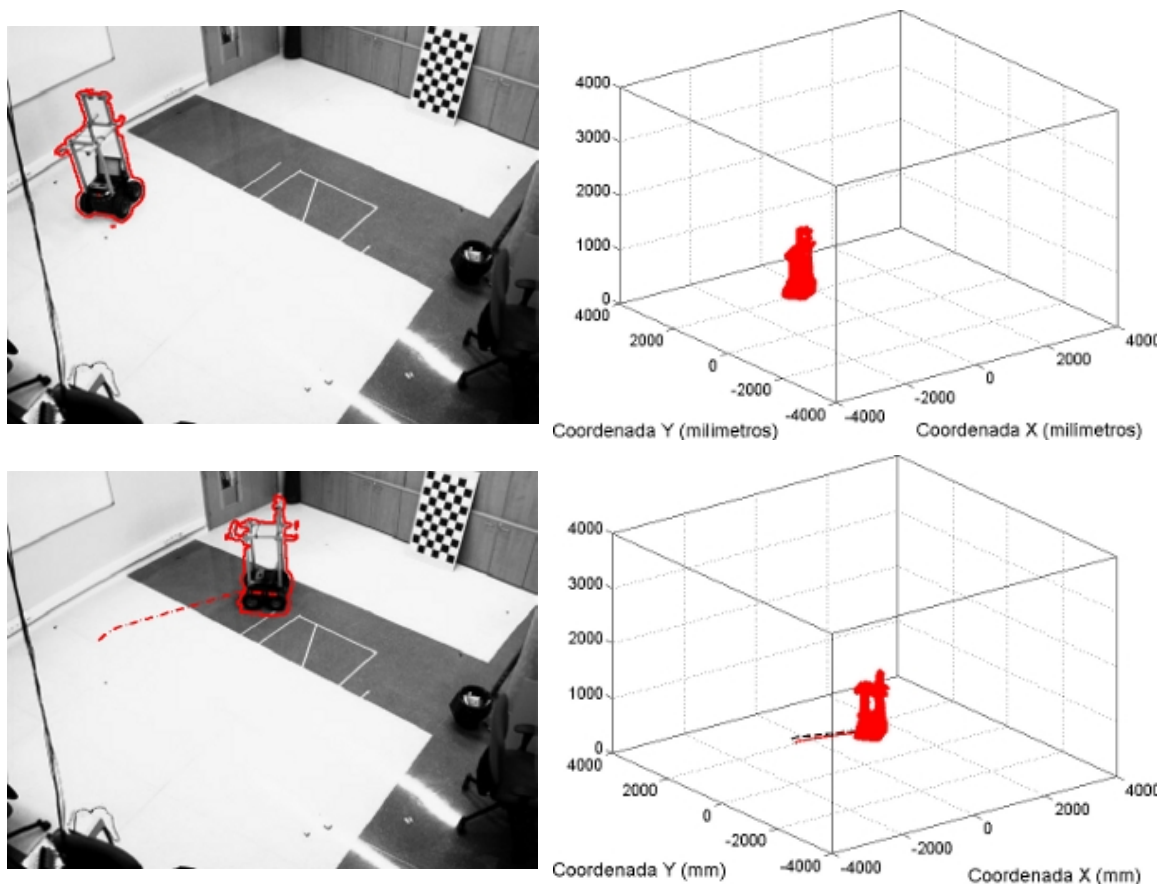


Figura 9.22. Curvas obtenidas de la segmentación de movimiento (columna izquierda) y resultado de la reconstrucción 3D mediante VH3D (columna derecha) para dos imágenes pertenecientes a la secuencia 1 (Figura 9.1). Además, se muestran la trayectoria real y la estimada.

En la Figura 9.23 se muestran los resultados para tres imágenes pertenecientes a la secuencia 2 (Figura 9.2). En esta figura se observa que, tras la segmentación de movimiento los dos robots, han sido separados del fondo y entre sí.



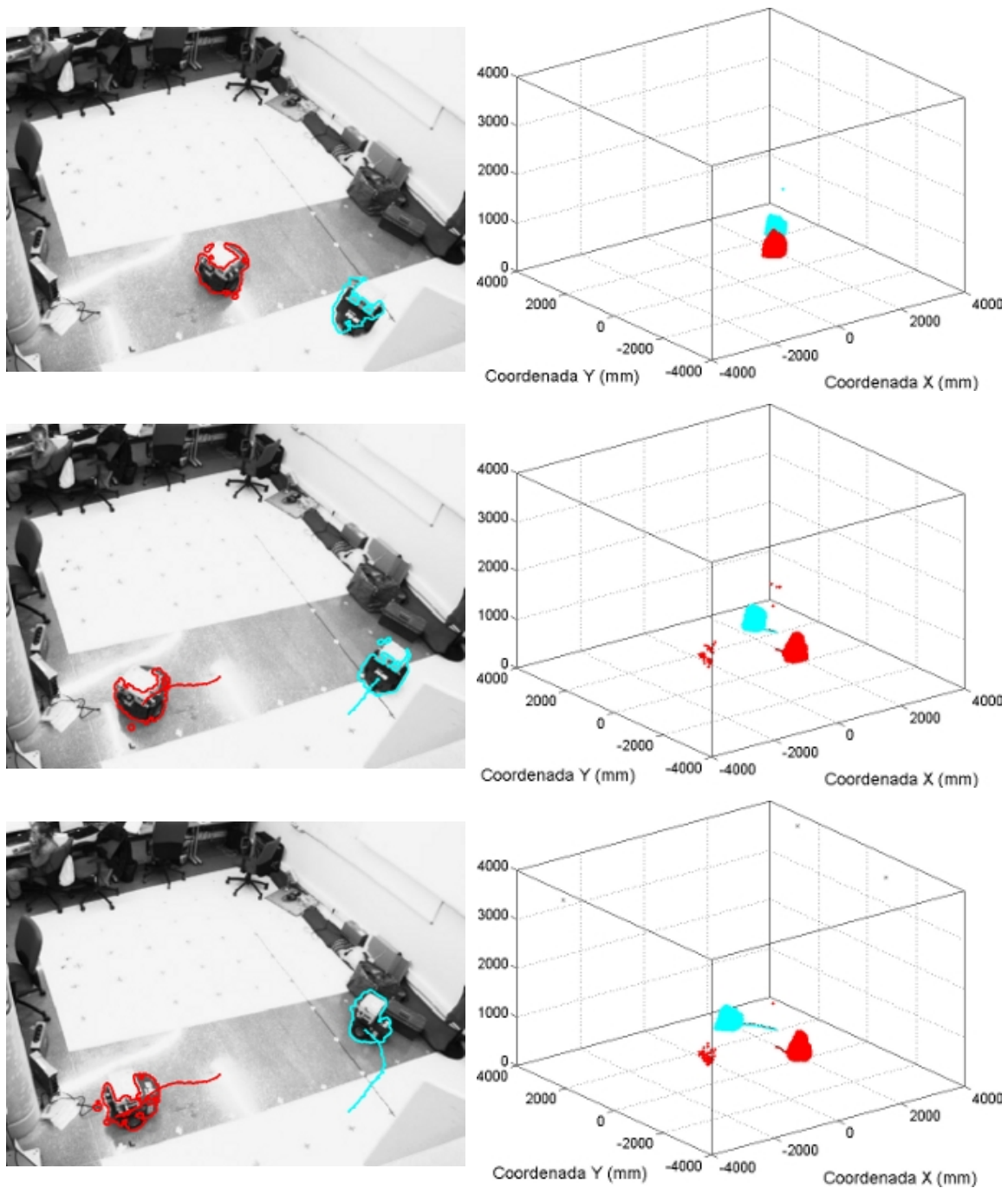


Figura 9.23. Curvas obtenidas de la segmentación de movimiento (columna izquierda) resultado de la reconstrucción 3D mediante VH3D (columna derecha) para tres imágenes pertenecientes a la secuencia 2 (Figura 9.2). En todas las imágenes se representa también la trayectoria estimada.

En la Figura 9.24 se muestran los resultados para tres imágenes que pertenecen a la secuencia 3 (Figura 9.3). Esta secuencia es muy similar a la secuencia 1, ya que se tiene un único robot en movimiento. Sin embargo, se ha decidido incluir ambas secuencias dado que en la secuencia 3 se cuenta con cambios importantes en la iluminación de la escena. De esta forma, el uso de la secuencia 3 permite comprobar la robustez del sistema frente a cambios de iluminación. En los resultados presentados en la Figura 9.24

se observa que el umbral dinámico propuesto funciona de forma correcta, permitiendo obtener las curvas incluso en presencia de cambios en la iluminación de la escena.

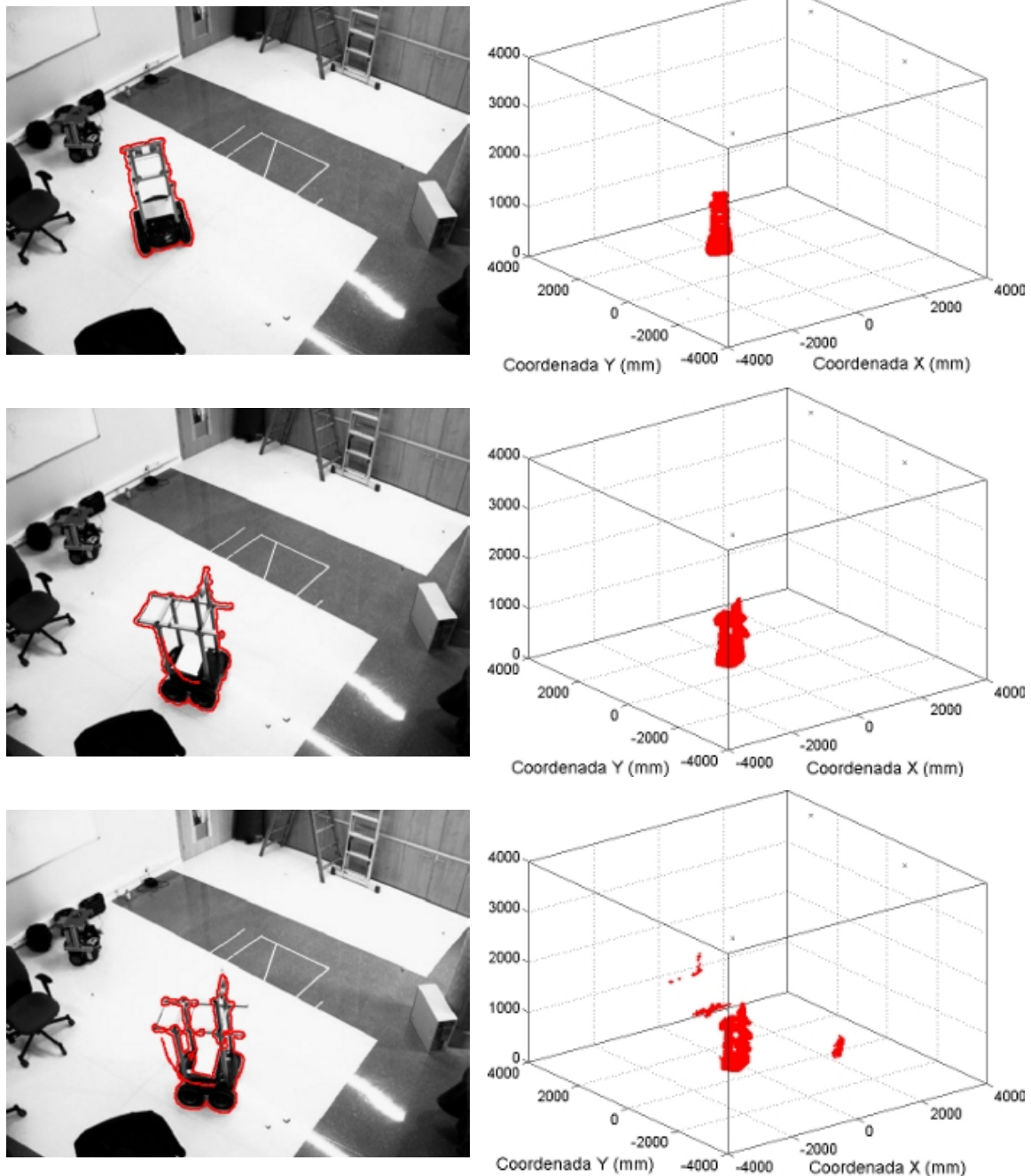


Figura 9.24. Curvas de la segmentación de movimiento (columna izquierda) resultado de la reconstrucción 3D mediante VH3D (columna derecha) obtenida a partir de los contornos de la segmentación para dos imágenes pertenecientes a la secuencia 3 (Figura 9.3).

Como ya se ha comentado a lo largo de esta memoria, la solución propuesta permite detectar y segmentar tanto robots como otros elementos móviles (usuarios, obstáculos, etc.). Esto se puede observar en las diferentes imágenes mostradas en la Figura 9.24, en las que se muestran los resultados obtenidos para tres imágenes que pertenecen a la secuencia 4 (Figura 9.4). En estas imágenes se incluye tanto un robot móvil (en color

azul en las figuras) como una persona (en rojo en las figuras) que se desplaza por el ISPACE-UAH. Se puede observar que, tanto el robot como la persona, se detectan de forma correcta. Conviene mencionar que la segmentación del robot es mejor que la de la persona. Esto se debe al hecho de que el algoritmo está planteado para elementos rígidos. A pesar de ello, en las imágenes de la Figura 9.24 se demuestra que también puede usarse para elementos que no sean rígidos, siempre que los cambios entre una imagen y la siguiente sean pequeños.

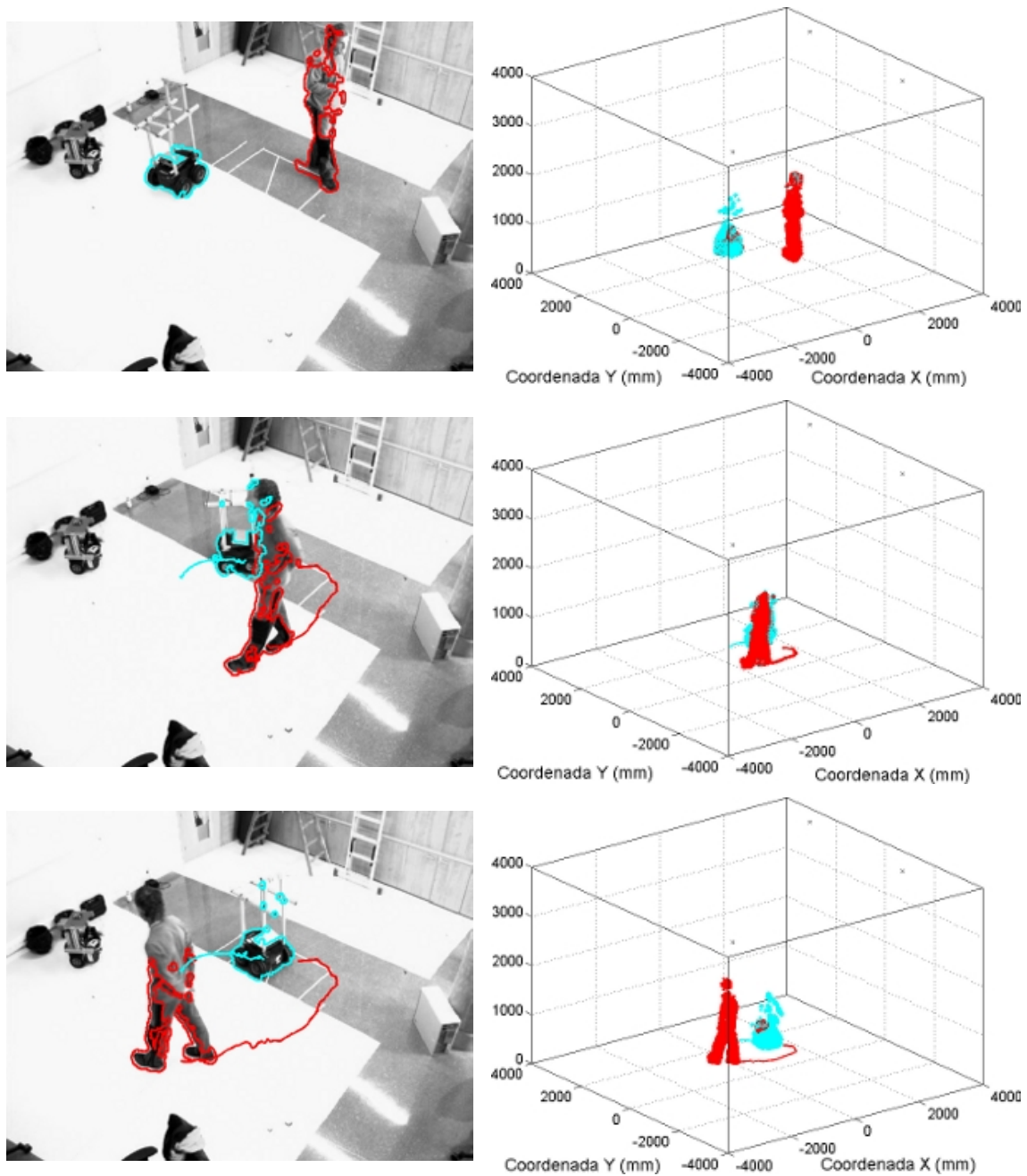


Figura 9.25. Curvas de la segmentación de movimiento (columna izquierda) resultado de la reconstrucción 3D mediante VH3D (columna derecha) obtenida a partir de los contornos de la segmentación para dos imágenes pertenecientes a la secuencia 4 (Figura 9.4).

La secuencia 5 (Figura 9.5) es la más compleja de todas las consideradas a lo largo de esta memoria. En ella se tiene un robot móvil que se desplaza por la escena, y un usuario que entra a la misma y deposita diferentes obstáculos (papeleras) sobre el suelo del ISPACE-UAH. En la Figura 9.26 se puede observar como, incluso en una secuencia compleja, ha sido posible realizar la segmentación de movimiento, y obtener la reconstrucción 3D de los diferentes elementos implicados.

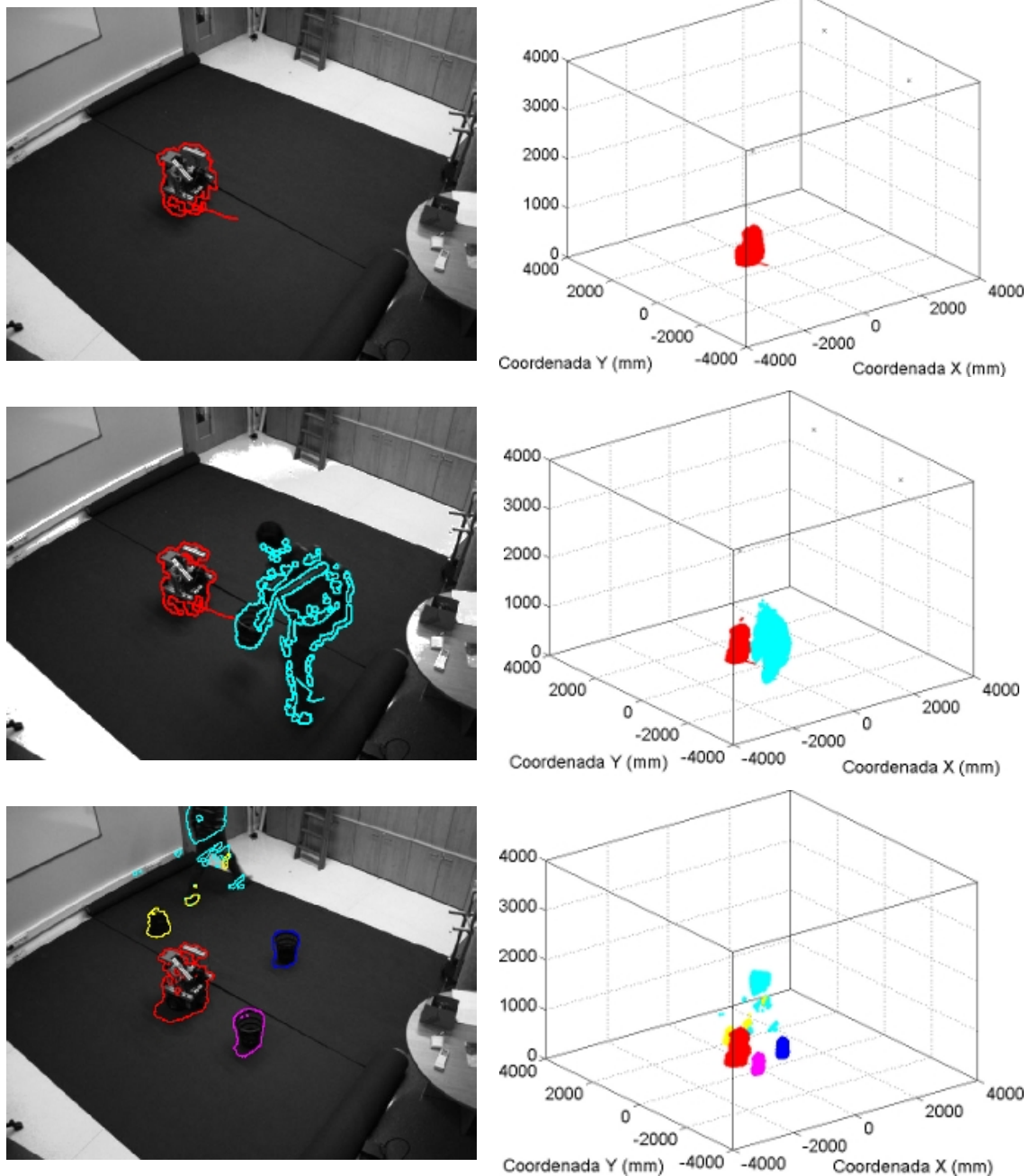


Figura 9.26. Curvas de segmentación de movimiento (columna izquierda) resultado de la reconstrucción 3D mediante VH3D (columna derecha) obtenida a partir de los contornos de la segmentación para dos imágenes pertenecientes a la secuencia 5 (Figura 9.5).



En la Figura 9.26 se observa de nuevo que la segmentación del usuario es menos precisa que la del robot móvil o los obstáculos. Esto se debe a la complejidad añadida por el hecho de que el usuario no pueda modelarse como un objeto rígido.

En las figuras anteriores se han mostrado los resultados obtenidos (segmentación y reconstrucción 3D) aplicando la solución propuesta en esta tesis a diferentes secuencias de imágenes. El tiempo consumido para la segmentación y posicionamiento es también un parámetro importante. En este sentido, en la Figura 9.27 se muestra el tiempo consumido para la obtención de la segmentación de movimiento de cada par de imágenes. Se puede observar que para las secuencias 1, 2 y 3, los valores de tiempo de procesamiento son siempre inferiores a 11 segundos para las imágenes consideradas. Además, el tiempo de procesamiento de la secuencia 2 es, en general, ligeramente superior al de las secuencias 1 y 3. Esto es debido a que en la secuencia 2 se tienen dos robots móviles, mientras que en 1 y 3 se cuenta con un único robot.

Por otro lado, en el caso de la secuencia 4, el tiempo se incrementa debido a que se realiza la segmentación de un robot y una persona. Dado que el algoritmo de segmentación se ha desarrollado bajo el supuesto de que los objetos móviles pueden modelarse como un sólido rígido, el hecho de incluir una persona (que no se comporta como un cuerpo rígido) incrementa la complejidad de la segmentación.

Finalmente, en la gráfica de tiempo de procesamiento de la secuencia 5 se distinguen dos partes. Las imágenes que corresponden a la primera parte (de la 1 a la 42) únicamente contienen un robot móvil. En el caso de las imágenes de la 43 a la 100 aparece una persona que va dejando obstáculos (papeleras) en diferentes posiciones de la escena. La aparición de la persona y los obstáculos se traducen en un incremento del tiempo necesario para obtener la segmentación de movimiento y la posición 3D.

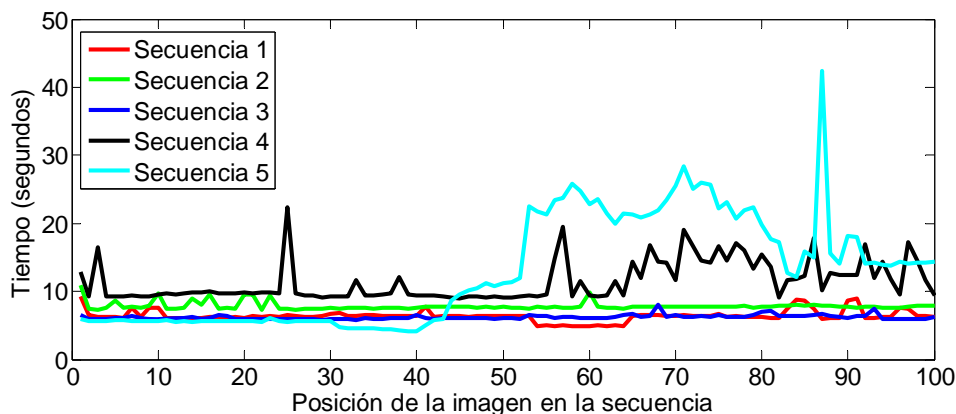


Figura 9.27. Tiempo de procesamiento de cada par de imágenes pertenecientes a cada una de las secuencias de test.

Conviene destacar que, aunque los valores de tiempo son elevados para una aplicación en tiempo real (del orden de varios segundos), estos tiempos son notablemente menores que los necesarios en caso de utilizar el algoritmo propuesto en [Sekkati & Mitiche 2006b], tal como se ha visto en apartados anteriores de este capítulo.

En la Figura 9.28 se muestra también el tiempo medio para cada una de las secuencias. Las conclusiones que se extraen de esta figura son similares a las presentadas para la Figura 9.27. Es sencillo apreciar que el tiempo de procesamiento se incrementa con la complejidad de la secuencia de imágenes. Así, el tiempo es más bajo para las secuencias 1 y 3, ya que únicamente se tiene un robot móvil. En caso de dos robots (secuencia 2), el tiempo medio se incrementa ligeramente. Por otro lado, en las secuencias en que aparecen usuarios, los tiempos se incrementan de forma notable, llegando a ser del doble que para el caso de un único robot.

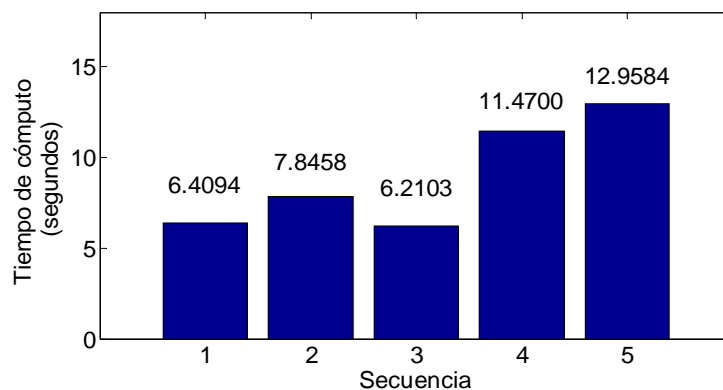


Figura 9.28. Valor medio del tiempo de cómputo consumido para la segmentación de movimiento y el posicionamiento 3D de múltiples robots móviles en cada una de las secuencias de test.

Respecto al error cometido en la estimación de la posición 3D, éste se calcula como la diferencia entre la posición estimada y la medida por los sensores odométricos de los robots (ecuación (9.1)). En la Figura 9.29 se muestra el valor del error de posicionamiento para las diferentes imágenes de las secuencias 1 a 4. Se puede observar que el peor caso se da para las imágenes pertenecientes a la secuencia 4 (Figura 9.29(d)). Pero, incluso en este caso, el valor del error es inferior a 250 milímetros para todas las imágenes consideradas. Es especialmente interesante el caso de la secuencia 2 (Figura 9.29(b)), en la que se observa que los dos robots se posicionan con un error inferior a 100 milímetros. Conviene recordar que, como ya se ha mencionado al inicio de este capítulo, las medidas de posición obtenidas a partir de los sensores odométricos de los robots utilizados en las secuencias 1 y 4 presentan un error de medida mayor que

los de los robots utilizados para las secuencias 2 y 3, debido a las diferentes características de adherencia de las ruedas (lo que hace que en algunos casos existan deslizamientos notables). Por este motivo, parte del error mostrado en las gráficas de la Figura 9.29(a) y (d) puede ser debido a esta circunstancia, siendo el error real de posicionamiento inferior al representado.

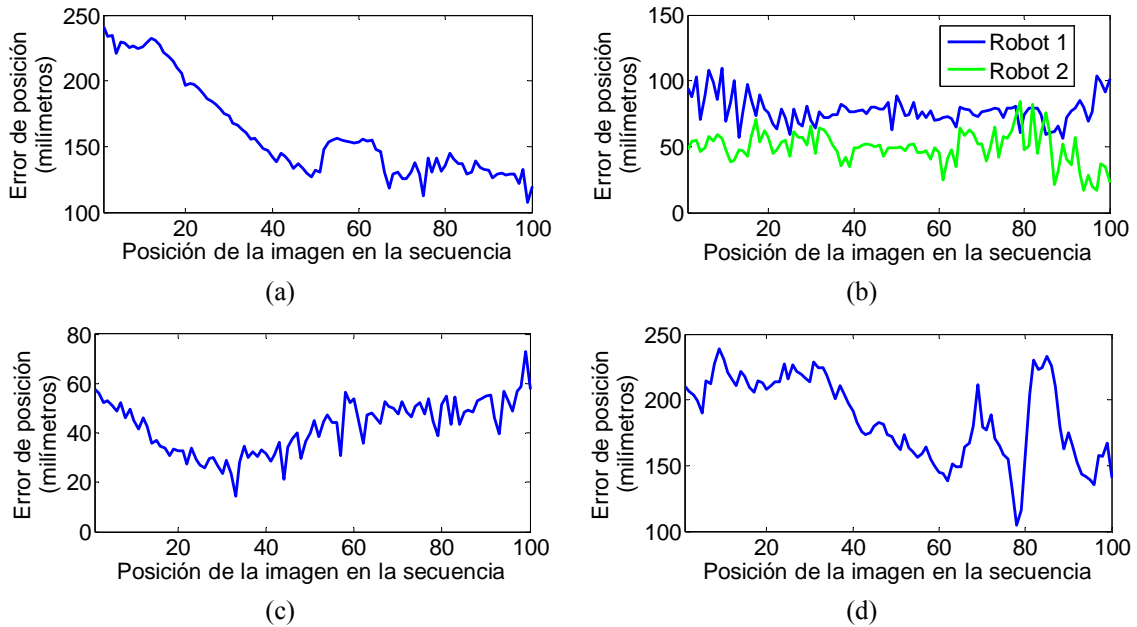


Figura 9.29. Error de posición calculado utilizando la ecuación (9.1) para cuatro secuencias de prueba. (a) Secuencia 1 (Figura 9.1). (b) Secuencia 2 (Figura 9.2). (c) Secuencia 3 (Figura 9.3). (d) Secuencia 4 (Figura 9.4).

Al igual que en el caso del tiempo de procesamiento, en la Figura 9.30 se representa el valor medio del error de posicionamiento para las secuencias 1, 2, 3 y 4.

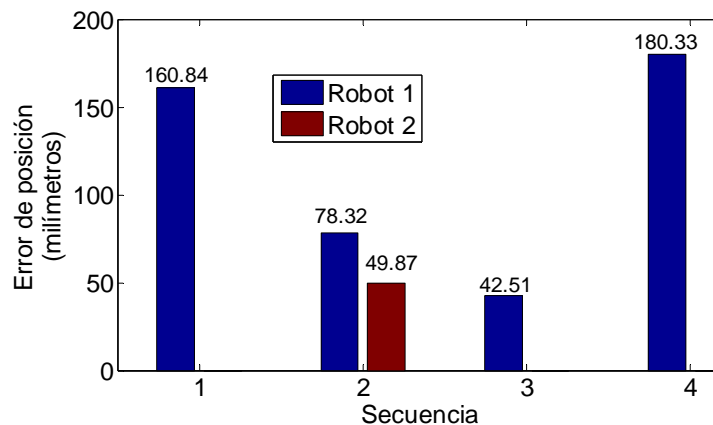


Figura 9.30. Valor medio del error de posición calculado utilizando la ecuación (9.1) para cada uno de los robots incluidos en las secuencias de test 1 (Figura 9.1), 2 (Figura 9.2), 3 (Figura 9.3) y 4 (Figura 9.4).

A modo de resumen, en la Tabla 9.7 se muestran los valores medios del tiempo de procesamiento y del error de posicionamiento obtenidos para 100 imágenes de cada una de las secuencias consideradas. En el caso de la secuencia 5 no se incluye el error de

posicionamiento debido a que no están disponibles las medidas de los sensores odométricos a bordo del robot.

Tiempo (s)	Secuencia 1	Secuencia 2	Secuencia 3	Secuencia 4	Secuencia 5
<b>Inicialización</b>	3.4305	3.3839	3.3206	4.7707	3.6549
<b>Minimización</b>	2.9441	4.6151	2.8579	7.3852	3.8811
<b>Filtrado</b>	0.0348	0.0341	0.0319	0.0374	0.0333
<b>Total</b>	<b>6.4094</b>	<b>8.0331</b>	<b>6.2103</b>	<b>12.1934</b>	<b>7.5693</b>
Error de Posición (mm)	Secuencia 1	Secuencia 2	Secuencia 3	Secuencia 4	Secuencia 5
<b>Robot 1</b>	160.8361	78.3235	42.5052	180.3307	-
<b>Robot 2</b>	-	49.8696	-	-	-

Tabla 9.7. Valor medio del tiempo de procesamiento de las diferentes etapas de la solución propuesta y error medio de posición. Resultados obtenidos para 100 imágenes pertenecientes a cada una de las secuencias de test.

Se muestra también la trayectoria estimada para cada uno de los robots de las secuencias 1 a 4, representada sobre el plano imagen, superpuesta a la imagen 100 de cada secuencia.

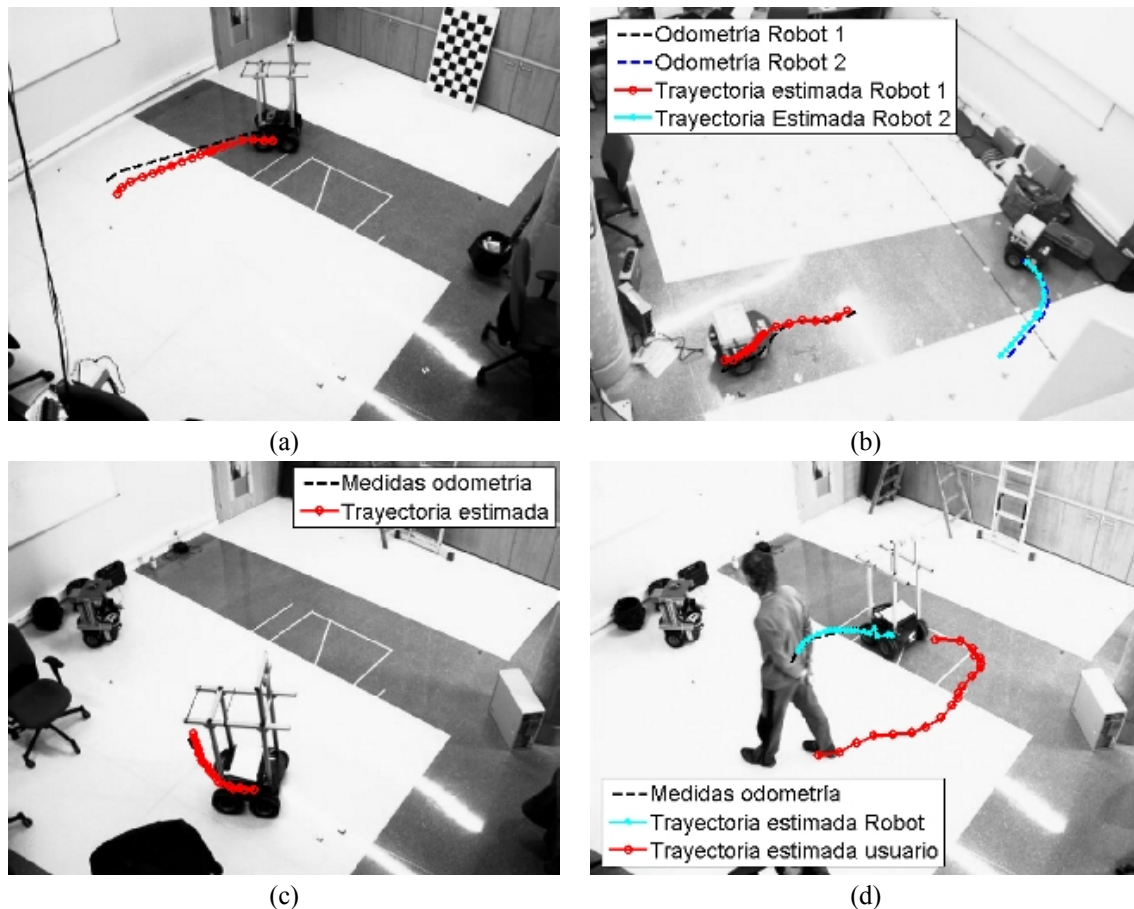


Figura 9.31. Trayectoria estimada representada sobre el plano imagen de la cámara 1 de las secuencias de prueba. (a) Secuencia 1 (Figura 9.1). (b) Secuencia 2 (Figura 9.2). (c) Secuencia 3 (Figura 9.3). (d) Secuencia 4 (Figura 9.4).



En la Figura 9.32 se representa la trayectoria sobre el plano del suelo (X,Y) en  $\Gamma_w$ . Además de la trayectoria, en las imágenes mostradas en la Figura 9.32 se ha incluido la reconstrucción 3D obtenida aplicando VH3D al resultado de la segmentación de movimiento de la imagen que ocupa la posición 100 de cada una de las secuencias consideradas.

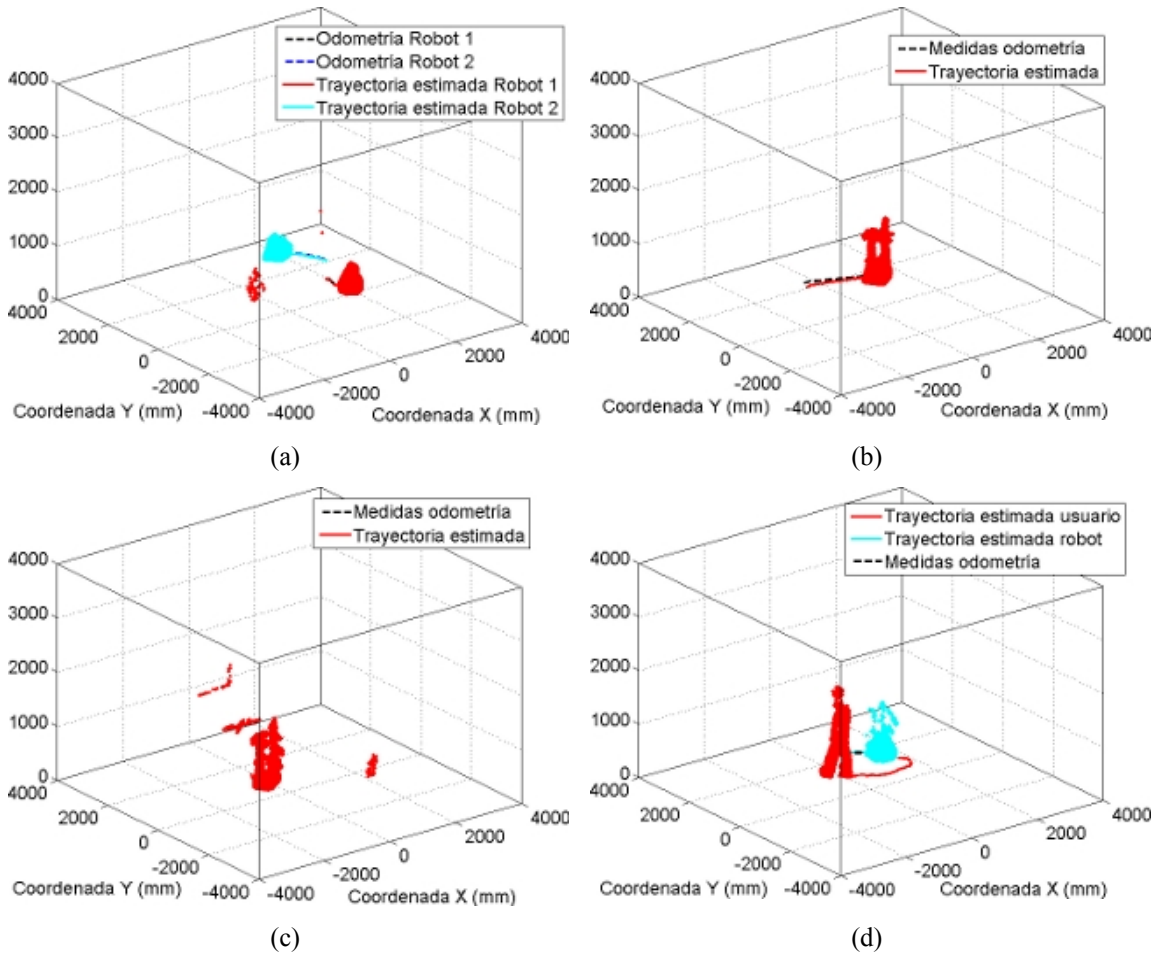


Figura 9.32. Trayectoria estimada a lo largo de cada secuencia y reconstrucción 3D de la imagen 100 de **las secuencias de prueba. (a) Secuencia 1** (Figura 9.1). **(b) Secuencia 2** (Figura 9.2). **(c) Secuencia 3** (Figura 9.3). (d) Secuencia 4 (Figura 9.4).

Finalmente, se representa la trayectoria estimada, junto con las medidas de los sensores odométricos de los robots proyectada sobre el plano del suelo  $X_w Y_w$  en  $\Gamma_w$ . En la Figura 9.33 se muestra la trayectoria estimada a partir de 500 imágenes pertenecientes a la secuencia 2 (Figura 9.2). Se ha elegido esta secuencia debido a que es la única que incluye dos robots móviles. En la gráfica mostrada en la Figura 9.33 se puede observar que las trayectorias estimadas se encuentran muy próximas a las medidas por los sensores odométricos de los robots móviles, a pesar de la complejidad de las trayectorias de los mismos.

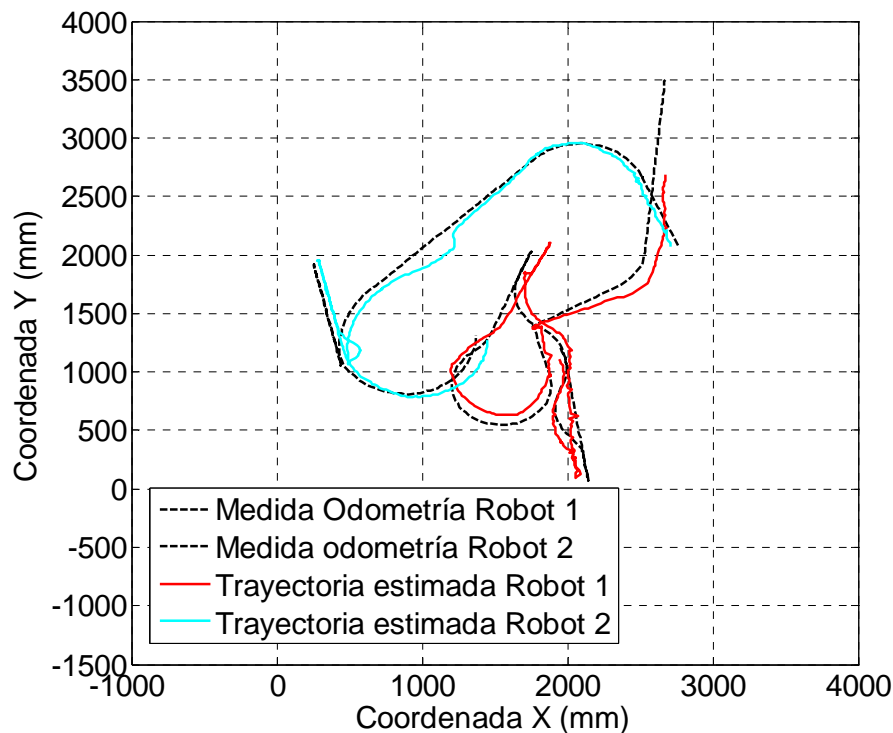


Figura 9.33. Trayectoria estimada a partir de 500 imágenes pertenecientes a la Secuencia 2 (Figura 9.2). Proyección sobre el plano (XY) en  $\Gamma_w$ .

## 9.6. Conclusiones

A la vista de los resultados obtenidos en las diversas pruebas experimentales realizadas, se puede concluir que las diferentes propuestas realizadas a lo largo de esta tesis son válidas para la segmentación de movimiento y el posicionamiento 3D de un número variable de robots móviles en espacios inteligentes.

Dada la diversidad de las propuestas realizadas a lo largo de esta tesis, las soluciones planteadas para cada una de las etapas han sido evaluadas de forma independiente. Esto ha permitido evaluar la incidencia de cada una de ellas sobre la solución final.

En resumen se puede concluir que la solución planteada en esta tesis, si bien es susceptible de que se introduzcan mejoras, permite realizar, de forma eficaz, la segmentación de movimiento y el posicionamiento 3D de robots móviles en espacios inteligentes, que era el objetivo fundamental de investigación perseguido en esta tesis.

## Capítulo 10

# **Resumen, conclusiones y aportaciones**



Tras describir cada una de las etapas que conforman la solución propuesta en esta tesis, en este capítulo final se resume dicha solución en su conjunto. Además, se realiza una recopilación de las conclusiones, líneas de trabajo futuro y aportaciones más relevantes de la tesis, así como de las publicaciones más relevantes derivadas de la misma.

## 10.1. Resumen

El objetivo de este resumen es presentar una visión conjunta de las diferentes soluciones que se han propuesto a lo largo de la tesis, y que conforman una solución global para la segmentación y posicionamiento de múltiples robots móviles en el ISPACE-UAH.

Para comenzar, en la Figura 10.1 se muestra un diagrama de bloques general del sistema propuesto en la tesis para la segmentación y posicionamiento de robots móviles.

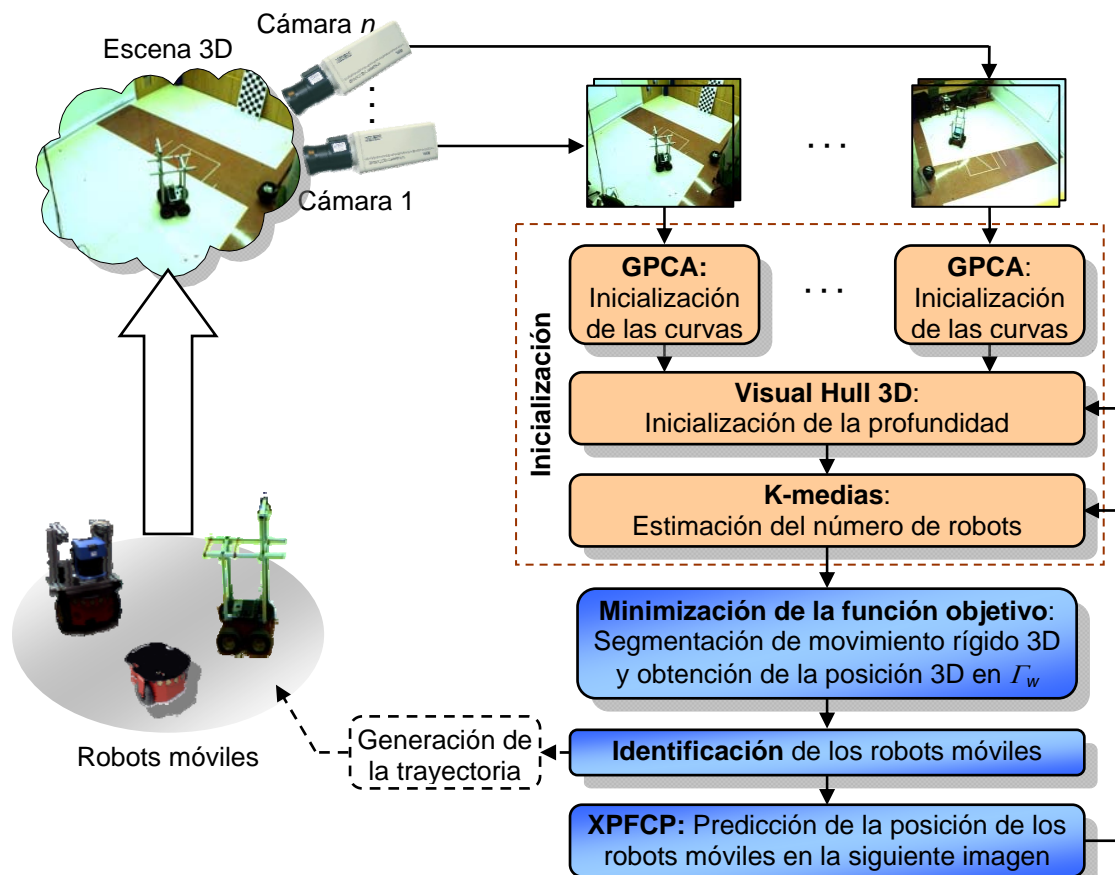


Figura 10.1. Diagrama de bloques general del algoritmo de segmentación de movimiento rígido y obtención de la posición 3D a partir de las imágenes captadas por  $n$  cámaras ubicadas en posiciones fijas del espacio.

En la Figura 10.1, las etapas implicadas en la inicialización de variables se muestran en color naranja, mientras que las relacionadas con la segmentación, posicionamiento y seguimiento de los robots se representan en color azul. Indicar que se han realizado numerosos experimentos prácticos, todos ellos a partir de imágenes tomadas en el ISPACE-UAH.

### 10.1.1. Inicialización de variables

La técnica de segmentación y posicionamiento planteada en esta tesis requiere la inicialización de un conjunto de variables. En concreto, las variables a inicializar son:

- Un conjunto de  $N-1$  curvas sobre el plano imagen de cada cámara. Estas curvas dividen la imagen en  $N$  fragmentos correspondientes a los  $N-1$  robots (u otros elementos móviles) y al fondo de la escena.
- La profundidad, entendida como la distancia entre cada punto de la escena y las cámaras (que coincide en este caso con la coordenada  $Z$  del punto en el sistema de referencia de la cámara  $\Gamma_c$ ).
- Estimación del número de robots móviles presentes en la escena.

Como se puede observar en el diagrama de bloques de la Figura 10.1, la inicialización de curvas se realiza de forma independiente para cada una de las cámaras, mientras que la profundidad inicial se obtiene conjuntamente para las  $n_c$  cámaras. Respecto al número de robots móviles, este se estima a partir de la rejilla de ocupación obtenida en la etapa de inicialización de la profundidad.

#### Inicialización de las curvas

Para la inicialización de las curvas se emplea el Análisis de Componentes Principales Generalizado (GPCA) propuesto en [Jieping et al. 2004]. La técnica GPCA se emplea tanto para el modelado del fondo de la escena como para la comparación de cada una de las imágenes de entrada con dicho modelo. El modelo de fondo se obtiene aplicando GPCA a un conjunto de  $N_i$  imágenes en las que no aparecen robots (ni otros elementos) móviles. La intensidad media de estas imágenes también se tiene en cuenta a la hora de comparar cada imagen de entrada con el modelo de fondo. Partiendo de que el fondo de la escena no cambia con el tiempo (el trabajo se desarrolla en un entorno

interior) el modelo de fondo puede obtenerse *off-line* ya que no requiere ser actualizado constantemente. La actualización del fondo es necesaria únicamente en caso de que se detecten cambios importantes en la iluminación de la escena. O bien, si el fondo de la escena se modifica.

Cada una de las imágenes de entrada se compara con el modelo de fondo. Esto se hace mediante la proyección al espacio transformado de GPCA y posterior recuperación de las mismas. La diferencia entre la imagen de entrada y la imagen recuperada será mayor en aquellos píxeles que pertenezcan a objetos que han entrado en la escena después del modelado del fondo. Por este motivo, aplicando un umbral sobre el error de recuperación es posible obtener los contornos de los robots móviles.

Para que el sistema sea robusto frente al ruido así como a los cambios en la iluminación, el umbral utilizado sobre el error de recuperación debe adaptarse a esos cambios. En este sentido se ha propuesto la utilización de un umbral adaptativo obtenido en función de la diferencia entre la intensidad media de la imagen de entrada y la de las  $N_i$  imágenes utilizadas en la etapa de modelado del fondo.

Las curvas obtenidas mediante GPCA se aproximan a los contornos reales de los robots. Sin embargo, debido a las condiciones de iluminación del entorno de movimiento de los robots (ISPACE-UAH, en nuestro caso), las sombras proyectadas por los robots sobre el plano del suelo también se incluyen en las curvas iniciales. Estas sombras se desplazan con la misma velocidad que los robots, por lo que no se eliminan en la etapa de segmentación, pudiendo provocar errores en el posicionamiento de los mismos. Con objeto de reducir estos errores se incluye en la inicialización de las curvas un proceso de eliminación de sombras de las curvas iniciales. Esta eliminación se lleva a cabo en un espacio invariante a la iluminación [Finlayson et al. 2002], [Finlayson et al. 2006].

Cabe destacar que el uso de GPCA para la inicialización de las curvas mejora sustancialmente la calidad de las curvas iniciales frente al uso de circunferencias que se plantea en los trabajos [Sekkati & Mitiche 2006a] ó [Sekkati & Mitiche 2006b], ya que las curvas iniciales se encuentran más próximas a los contornos reales de los robots. Esto se traduce en una reducción del número de iteraciones del algoritmo de minimización de la función objetivo, con la consiguiente disminución del tiempo consumido por el mismo.

### **Inicialización de la profundidad**

La inicialización de la profundidad se realiza a partir de las curvas iniciales obtenidas para cada una de las cámaras. A partir de esas curvas, se utiliza Visual Hull 3D (VH3D) para obtener una rejilla de ocupación en el sistema de referencia global,  $\Gamma_w$ , que define los contornos de los robots móviles en 3D.

Debido a que el VH3D es costoso computacionalmente, en la tesis se ha propuesto incluir un seguidor a la salida del proceso de segmentación y posicionamiento. La información de salida del mismo se realimenta a la etapa de inicialización con el objetivo de reducir, en la medida de lo posible, el tiempo de procesamiento.

Para el seguimiento de los robots móviles se ha elegido un filtro de partículas extendido con proceso de clasificación (XPFCP). La elección de este filtro de partículas frente a otras alternativas se debe a que, dada la multimodalidad del mismo, permite realizar el seguimiento de múltiples robots móviles con un único filtro. Además, el incremento del número de robots a seguir no aumenta el tiempo consumido por el XPFCP.

### **Estimación del número de robots y separación de las curvas iniciales**

El algoritmo de segmentación y posicionamiento 3D propuesto requiere conocimiento a priori del número de robots móviles presentes en la escena. Por este motivo se incluye una etapa que permite estimar el número de robots a partir de las posiciones ocupadas en la rejilla obtenida mediante VH3D.

La estimación del número de robots (y otros objetos móviles) se lleva a cabo usando una técnica de clasificación basada en el algoritmo  $k$ -medias extendido. Esta técnica requiere la elección de un valor de distancia que determina cuándo se crea una nueva clase. Este valor, denominado distancia máxima ( $dMax$ ) se ha determinado en función del tamaño aproximado de los robots a detectar. El valor elegido se ha validado mediante un conjunto de pruebas experimentales. Tras la clasificación se obtiene una estimación del número de robots móviles que se emplea en la etapa de segmentación y posicionamiento. Además, los puntos 3D correspondientes a las posiciones ocupadas se clasifican en función del elemento al que pertenecen. Esto permite dividir las curvas iniciales obtenidas mediante GPCA, obteniéndose un conjunto de curvas, identificadas en función del robot (o elemento móvil) con el que se corresponden.



### 10.1.2. Segmentación de movimiento y posicionamiento 3D

En esta tesis, tanto la segmentación de movimiento, como la posición 3D de los robots móviles se obtienen mediante la minimización de una función objetivo. Esta función objetivo se define en función de diferentes variables relacionadas con la posición y la velocidad de los robots, así como con los parámetros de las cámaras.

Una de las aportaciones más importantes de esta tesis es la obtención de una función objetivo que incluye la información de múltiples cámaras. Esto ha permitido incrementar la robustez de la solución propuesta, así como mejorar los resultados de segmentación y posicionamiento frente al uso de una función por cada una de las cámaras [Sekkati & Mitiche 2006b].

A partir de los valores iniciales de las curvas y la profundidad, así como el número de robots móviles estimado en la etapa de inicialización, la minimización de la función objetivo se lleva a cabo mediante un algoritmo *greedy* formado por tres etapas que se repiten hasta la convergencia del mismo. Tras la minimización se obtienen los siguientes resultados:

- Un conjunto de  $N-1$  curvas sobre cada imagen que se ajustan a los contornos de los  $N-1$  robots móviles,
- Las componentes de velocidad lineal y angular de cada uno de los robots. Estas velocidades son relativas al sistema de referencia global  $\Gamma_w$ , por lo que son las mismas para todas las cámaras.
- La profundidad de cada uno de los puntos que pertenecen a los robots móviles en el sistema de referencia de cada cámara  $\{Z_{ci}\}_{i=1}^{n_c}$ . Este valor permite obtener una reconstrucción en 3D de los robots.

Tanto el número de iteraciones hasta la convergencia, como la calidad de los resultados (curvas, velocidades y profundidad) obtenidos tras la minimización de la función objetivo dependen fuertemente de los valores iniciales de las variables y de las constantes que ponderan la contribución de cada uno de los términos en la función objetivo. Por este motivo, se ha realizado un estudio exhaustivo que permite determinar cuáles son los valores más adecuados para esas constantes.

### 10.1.3. Identificación y seguimiento de los robots móviles

Tras la segmentación de movimiento y la estimación de la posición 3D de todos los elementos móviles presentes en la escena, la identificación de los robots se lleva a cabo mediante la comparación de las componentes de la velocidad lineal en  $X_w$  e  $Y_w$  estimadas por el algoritmo de segmentación de movimiento y las medidas de los sensores odométricos a bordo de los mismos. Esto es posible porque los robots son agentes controlados por el sistema de supervisión del espacio inteligente.

Para el seguimiento de los robots se emplea un filtro de partículas extendido con proceso de clasificación (XPFCP). El XPFCP permite realizar el seguimiento de un número variable de robots con un único estimador, sin necesidad de aumentar el vector de estado.

La información obtenida a la salida del XPFCP se realimenta a la etapa de inicialización de variables, ya que supone una buena estimación de la posición que ocuparán los robots móviles en la siguiente imagen de la secuencia. Esta realimentación permite reducir el tiempo de cómputo consumido en la etapa de inicialización de variables.

Las diferentes propuestas realizadas a lo largo de esta tesis han sido validadas de forma experimental. Para ello se han utilizado diferentes secuencias de imágenes adquiridas en el ISPACE-UAH.

## 10.2. Conclusiones

En esta tesis se ha presentado una solución que permite la segmentación de movimiento y la estimación de la posición 3D de múltiples robots móviles en un espacio dotado de un array de cámaras, sincronizadas y calibradas, ubicadas en posiciones fijas del entorno (espacio inteligente). En la solución propuesta, tanto la segmentación de movimiento, como la estimación de la posición 3D se obtienen mediante la minimización de una función objetivo. Esta función incorpora información relacionada con el campo de movimiento sobre el plano imagen, así como con el movimiento 3D de los robots móviles. Este tipo de técnicas presenta algunas ventajas frente a otras propuestas, ya que no requiere conocimiento a priori de las características

de los robots. Tampoco es necesario el uso de marcas artificiales o balizas. Por otro lado, permite obtener la segmentación de movimiento y estimar la posición 3D de los robots de forma concurrente, en lugar de secuencial.

La segmentación de movimiento y el posicionamiento 3D se llevan a cabo a partir de una función objetivo, utilizando la información de todas las cámaras que ven a los robots, planteándose dos alternativas.

En la primera alternativa, se obtiene la segmentación y posición 3D relativa a cada una de las cámaras de forma independiente. Posteriormente, se incluye un proceso de fusión que permite incorporar la información de todas las cámaras. De esta forma se obtiene un resultado global. Sin embargo, en la minimización de la función objetivo, que se realiza para cada cámara, únicamente se tiene en cuenta la información de cada una de ellas. Por este motivo, los resultados obtenidos empleando esta alternativa son poco robustos.

Con objeto de incrementar la robustez y mejorar la precisión de los resultados de la segmentación y posicionamiento 3D se ha propuesto una nueva función objetivo que incorpora la información adquirida por múltiples cámaras. La minimización de esta función permite obtener la segmentación de movimiento y estimar la posición 3D de múltiples robots móviles teniendo en cuenta la información obtenida de forma simultánea por todas las cámaras disponibles.

En la presente tesis se ha puesto de manifiesto la importancia de la correcta inicialización de variables (curvas que definen los contornos de la segmentación y profundidad en cada punto de la imagen). La importancia de la inicialización radica en la fuerte dependencia que presenta tanto el tiempo de procesamiento, como los resultados de la segmentación y la estimación de la posición 3D respecto a los valores iniciales de las variables implicadas.

El proceso propuesto para la inicialización de las curvas consta de tres partes bien diferenciadas. Las dos primeras están basadas en GPCA y consisten en la obtención de un modelo del fondo de la escena y la comparación de cada imagen de entrada con dicho modelo. La comparación se lleva a cabo mediante la umbralización del error de recuperación, definido como la diferencia entre la imagen original y la imagen obtenida tras la proyección al espacio transformado GPCA y posterior recuperación de la misma. Tras las dos primeras etapas se obtiene un conjunto de curvas que se aproximan a los

contornos reales de los robots móviles. Sin embargo, debido a las condiciones de iluminación, es frecuente que los robots proyecten sombras sobre el plano del suelo, que se incluyen en las curvas iniciales. Estas sombras no se eliminan durante la segmentación de movimiento debido a que se desplazan a la misma velocidad que los robots. Por este motivo, en la inicialización de las curvas se ha incluido una tercera etapa en la que se eliminan las sombras de las curvas iniciales. La eliminación de sombras se realiza tras proyectar las imágenes a un espacio invariante a la iluminación.

También se ha propuesto una solución para la inicialización de la profundidad en aquellos casos en los que se disponga de imágenes de múltiples cámaras. La técnica propuesta está basada en el uso de Visual Hull 3D (VH3D) para relacionar la información de las  $n_c$  cámaras disponibles. Esto permite obtener una rejilla de ocupación 3D del espacio, a partir de la cual se obtiene una buena estimación de los valores iniciales de la profundidad.

Por otro lado, la eliminación de las muestras en el sistema de referencia global  $\Gamma_w$  para las que la altura es cero permite reducir las sombras proyectadas por los robots sobre el plano del suelo sin necesidad de transformar las imágenes a un espacio invariante a la iluminación. Esta alternativa es espacialmente útil en los casos en que el número de cámaras que captan simultáneamente los robots sea elevado, por las ventajas que ello supone desde el punto de vista de segmentación, sin suponer un incremento notable del tiempo de cómputo.

Así mismo, se han presentado dos alternativas que permiten la estimación del número de robots móviles presentes en la escena. Ambas alternativas están basadas en una versión extendida del algoritmo de clasificación  $k$ -medias. En el primer caso, este algoritmo se aplica sobre los puntos interiores a las curvas sobre el plano imagen. Esta alternativa es especialmente útil para aplicaciones en las que únicamente se dispone de información visual de una cámara. En el caso de tener múltiples cámaras, el uso de  $k$ -medias extendido para la clasificación del conjunto de puntos 3D, obtenidos tras la inicialización de la profundidad mediante VH3D, proporciona resultados más precisos.

Los valores de las constantes que ponderan la contribución de cada uno de los términos de las funciones objetivo, así como la función de preservación de bordes elegida, también influyen de forma notable en los resultados obtenidos. Por este motivo, en la tesis se ha realizado un estudio exhaustivo de diferentes funciones de preservación

de bordes presentes en la literatura. En el estudio realizado se ha corroborado que cualquiera de las funciones de preservación de bordes presentadas es válida para la aplicación planteada en esta tesis. Esto es así siempre que los valores de las constantes de ponderación de los diferentes términos de la función objetivo se ajusten de forma adecuada.

Para el caso de la función cuadrática (elegida por su simplicidad y porque los resultados no difieren mucho con respecto a otras alternativas) se han obtenido los valores más adecuados para las constantes de ponderación de la función objetivo, tanto para el caso de una como para múltiples cámaras.

En ambos casos, los valores elegidos han sido validados con diferentes pruebas experimentales, contemplando diferentes condiciones ambientales y de presencia de robots y otros objetos. Cabe destacar aquí que los valores de las constantes ajustados para una secuencia de imágenes han sido utilizados al procesar el resto de secuencias, siendo válidos en todos los casos estudiados.

Tras la segmentación de movimiento es necesario realizar la identificación de los robots móviles. Esta identificación se lleva a cabo mediante la comparación de las componentes de la velocidad lineal en  $X_w$  e  $Y_w$  estimadas por el algoritmo de segmentación de movimiento y las medidas de los sensores odométricos a bordo de los mismos. Esto es posible porque los robots son agentes controlados por el sistema de supervisión del espacio inteligente. Además de identificar los diferentes robots, en esta etapa también es posible diferenciar los robots de los usuarios u obstáculos presentes en la escena.

Tras la identificación se ha incluido una etapa de seguimiento basada en un filtro de partículas extendido con proceso de clasificación (XPFCP). El XPFCP permite realizar el seguimiento de un número variable de robots con un único estimador, sin necesidad de aumentar el vector de estado.

La información obtenida a la salida del XPFCP se realimenta a la etapa de inicialización de variables, ya que supone una buena estimación de la posición que ocuparán los robots móviles en la siguiente imagen de la secuencia.

Todas las propuestas realizadas en la tesis han sido validadas de forma experimental. Para ello se han realizado diversas pruebas experimentales sobre secuencias de imágenes adquiridas en el ISPACE-UAH. En las secuencias consideradas

se ha intentado representar las diferentes situaciones que pueden darse en el ISPACE-UAH. Así, se incluyen secuencias de imágenes con cambios de iluminación, así como secuencias en las que, además de los robots móviles, aparecen usuarios y otros obstáculos.

### **10.3. Aportaciones más relevantes**

Si bien dentro de cada uno de los capítulos que conforman esta memoria se han enumerando las diferentes aportaciones realizadas en la tesis, a continuación se realiza una recopilación de las más relevantes.

Entre las aportaciones más notables se encuentra la propuesta de una función objetivo para el caso de múltiples cámaras, así como el algoritmo de minimización de la misma.

También es destacable la propuesta de una solución que permite realizar la fusión de los resultados en caso de que la minimización de la función objetivo se ejecute de forma independiente por cada una de las cámaras. Por otra parte, se ha definido un criterio de convergencia del proceso de minimización para la aplicación de interés de esta tesis. Además, este criterio es el mismo para las dos soluciones propuestas, permitiendo comparar los resultados obtenidos en ambos casos.

Dada la importancia de la correcta inicialización de las variables implicadas en la función objetivo, una parte importante de los esfuerzos realizados se ha dedicado a la búsqueda de alternativas de inicialización. En esta línea se ha propuesto una solución, basada en GPCA, que permite obtener un conjunto de curvas iniciales que se aproximan a los contornos reales de los robots móviles. Además, se ha propuesto un umbral dinámico en función de la diferencia de la intensidad media de la imagen de entrada y la de las imágenes utilizadas para la obtención del modelo de fondo de la escena. El uso de este umbral dinámico incrementa, de forma notable, la robustez del proceso de inicialización de las curvas frente a cambios en la iluminación de las imágenes.

Respecto a la inicialización de la profundidad, el uso de Visual Hull 3D ha permitido relacionar la información de todas las cámaras disponibles en el sistema de coordenadas del mundo  $\Gamma_w$ , proporcionando una buena estimación de la profundidad inicial.

Por otro lado, se ha propuesto una solución basada en una versión extendida del algoritmo de clasificación  $k$ -medias para la estimación del número de robots móviles. Las diferentes aportaciones realizadas en el proceso de inicialización de variables permiten obtener la segmentación de movimiento y la posición 3D de los robots en un número menor de iteraciones del algoritmo de minimización de la función objetivo, con la consiguiente reducción del tiempo de procesamiento. Además, incrementan la robustez del sistema frente a cambios en la iluminación.

Por otro lado, se ha comprobado que los resultados de la segmentación y la estimación de la posición 3D también presentan una dependencia notable frente a la función de preservación de bordes elegida, y los valores de las constantes que ponderan la contribución de cada uno de los términos implicados en la función objetivo. En esta línea se ha realizado un estudio exhaustivo de diferentes funciones de preservación de bordes que existen en la literatura, permitiendo concluir que todas ellas son igualmente válidas, siempre que los valores de las constantes se ajusten de forma correcta. El proceso de ajuste de las constantes para la aplicación de interés es otra de las aportaciones realizadas.

Tras la minimización de la función objetivo se ha incorporado un XPFCP para el seguimiento de los robots móviles. La información obtenida a la salida del XPFCP se realimenta a la etapa de inicialización de variables, ya que supone una buena estimación de la posición que ocuparán los robots móviles en la siguiente imagen de la secuencia. Esta propuesta de realimentación supone otra de las aportaciones de esta tesis. La importancia de la misma radica en que permite reducir el tiempo consumido en la inicialización.

Los resultados obtenidos en las diferentes pruebas experimentales realizadas han permitido concluir que las diferentes propuestas realizadas a lo largo de esta tesis son de aplicación para la segmentación de movimiento y el posicionamiento 3D de un número variable de robots móviles en espacios inteligentes, cumpliéndose así con los objetivos planteados en la tesis.

## 10.4. Líneas de trabajo futuro

Si bien en esta tesis se han propuesto soluciones al tema de localización y seguimiento de robots móviles en un espacio inteligente utilizando cámaras ubicadas en puntos fijos del espacio, es evidente que existen problemas aún por resolver y aspectos planteados en esta tesis que se pueden mejorar. En este sentido, seguidamente se enumeran algunas líneas de trabajo futuro que pudieran ser de interés:

- Una de estas tareas, quizás la más inmediata, es la implementación en tiempo real de las diferentes soluciones propuestas a lo largo de la tesis.
- La solución propuesta para la segmentación de movimiento y el posicionamiento 3D está restringida al caso de objetos rígidos. En este sentido, otra línea de trabajo futuro interesante es la investigación de soluciones para la segmentación y posicionamiento de objetos deformables.
- Las diferentes propuestas presentadas en la tesis han sido evaluadas en un único espacio (ISPACE-UAH). En este sentido, una línea futura de investigación sería trasladar la solución propuesta a espacios más amplios y con múltiples estancias.
- Finalmente, también se considera interesante la posibilidad de reemplazar las cámaras utilizadas en nuestra propuesta por cámaras de tiempo de vuelo (TOF, Time-Of-Flight). Este tipo de cámaras permite adquirir tanto imágenes en blanco y negro de la escena como la distancia a cada uno de los puntos de la escena captada (utilizando un sistema de iluminación infrarrojo, integrado en la propia estructura de la cámara). Ello podría resultar de interés, tanto desde el punto de vista de reducir el número de cámaras a ubicar en el entorno, como de simplificar los algoritmos para la segmentación y posicionamiento 3D de los robots.



## 10.5. Publicaciones derivadas de la tesis

Seguidamente se enumeran algunas de las publicaciones más relevantes que se han derivado, hasta el momento, de esta tesis.

- Cristina Losada, Manuel Mazo, Sira Palazuelos, Daniel Pizarro, Marta Marrón. Multi-camera sensor system for 3D segmentation and localization of multiple mobile robots. *Sensors* (Accepted for publication)
- Daniel Pizarro, Manuel Mazo, Enrique Santiso, Marta Marron, David Jimenez, Santiago Cobreces and Cristina Losada. Localization of Mobile Robots using Odometry and an External Vision Sensor. *Sensors*. (Accepted for publication).
- C. Losada, M. Mazo, S. Palazuelos, J. L. Martín, J. J. García. Motion segmentation using GPCA techniques and optical flow. Euro American Conference on Telematics and Information Systems 2007 (EATIS 2007). EATIS'07 ACM-DL Proceedings (ISBN: 978-1-59593-598-4). 2007.
- C. Losada, M. Mazo, S. Palazuelos, D. Pizarro, M. Marrón, F. Redondo. 3D Motion Segmentation and 3D Localization of Mobile Robots Using an Array of Static Cameras and Objective Function Minimization. Proc. of the 2009 IEEE International Symposium on Intelligent Signal Processing. WISP 2009. (ISBN: 978-1-4244-5059-6). 2009.
- C. Losada, M. Mazo, S. Palazuelos, F. Redondo. Adaptive Threshold for Robust Segmentation of Mobile Robots from Visual Information of their own Movement. Proc. of the 2009 IEEE International Symposium on Intelligent Signal Processing. WISP 2009. (ISBN: 978-1-4244-5059-6). 2009.
- Cristina Losada, Manuel Mazo, Sira Palazuelos, Daniel Pizarro, Marta Marron. Motion Segmentation and 3D Positioning of Multiple Mobile Robots Using an Array of Static Cameras in an Intelligent Space. IEEE International Symposium on Industrial Electronics 2010. ISIE 2010. (Accepted )
- Cristina Losada, Juan Jesús García, Manuel Mazo, Jesús Ureña, Álvaro Hernández, M<sup>a</sup> Jesús Díaz, Carlos De Marziani. Uso de la técnica PCA para la validación de la detección de objetos en entornos ferroviarios. Actas del XIII Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI'06). ISBN: 84-8317-564-9 (Edición en CD). 2006.

- Cristina Losada, Manuel Mazo, Sira Palazuelos. Segmentación de objetos en movimiento utilizando GPCA, técnicas algebraicas y flujo óptico para aplicaciones en espacios inteligentes. Actas del XIV Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI'07). 2007.
- Cristina Losada, Manuel Mazo, Sira Palazuelos, Francisco Redondo. Segmentación robots móviles en espacios inteligentes utilizando técnicas GPCA y minimización de funciones de energía. Actas del XV Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI'08). ISBN: 13:978-84-96997-04-2. 2008.
- Losada, Cristina; Mazo, Manuel; Palazuelos, Sira; Blanco, Edward. Segmentación y posicionamiento de sillas de ruedas en espacios inteligentes mediante minimización de funciones de energía. Memorias del V Congreso de tecnologías de apoyo a la discapacidad (IBERDISCAP 2008). ISBN: 978-958-8316-63-5. pp: 271-274. 2008.

# Referencias

[Alsabti et al. 1998]

Alsabti, K.; Ranka, S.; Singh, V. *An efficient k-means clustering algorithm*. Proc. of the First Workshop on High Performance Data Mining at the International Parallel Processing Symposium (IPPS98). 1998.

[Aubert et al. 1999]

Aubert, G.; Deriche, R. and Kornprobst, P. *Computing optical flow via variational techniques*. SIAM Journal on Applied Mathematics 1999. Vol. 60, Issue 1, pp. 156-182. 1999.

[Black & Anandan 1991]

Black, M. and Anandan, P. *Robust dynamic motion estimation over time*. Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition. Pp: 296-302. 1991.

[Boult & Brown 1991]

Boult, T.E. and Brown, L.G. *Factorization-based segmentation of motions*. Proc. of the IEEE Workshop on Motion Understanding. 1991. Pages 179-186. 1991.

[Briggs et al. 2000]

Briggs, A.J.; Scharstein, D.; Braziunas, D.; Dima, C.; Wall, P. *Mobile robot navigation using self-similar landmarks*. Proc. of the IEEE International Conference on Robotics and Automation, 2000. ICRA '00. Vol.2, pp.1428-1434. 2000.

[Brooks 1997]

Brooks, R. A. *The Intelligent Room project*. Proc. of the 2nd international Conference on Cognitive Technology (CT '97) 1997.

[Bršćic & Hashimoto 2008]

Bršćic', D.; Hashimoto, H. *Mobile Robot as Physical Agent of Intelligent Space*. Journal of Computing and Information Technology (1330-1136). Vol. 17, issue. 1, pp. 81-94. 2008.

[Busso et al. 2005]

Busso, C.; Hernanz, S.; Chi-Wei Chu; Soon-il Kwon; Sung Lee; Georgiou, P.G.; Cohen, I.; Narayanan, S. *Smart room: participant and speaker localization and identification*. Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. (ICASSP '05). Vol.2, pp. ii/1117-ii/1120. 2005.

[Busso et al. 2007]

Busso, C.; Georgiou, P.G.; Narayanan, S.S. *Real-Time Monitoring of Participants' Interaction in a Meeting using Audio-Visual Sensors*. Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. Vol.2, pp.II-685-II-688. 2007.

[Coelho & Campos 1999]

Coelho, L.S.; Campos, M.F.M. *Pose estimation of autonomous dirigibles using artificial landmarks*. Proc. of the XII Brazilian Symposium on Computer Graphics and Image Processing, 1999. pp.161-170. 1999

[Coen 1998]

Coen, M. H. *Design principles for intelligent environments*. Proc. of the Fifteenth National/Tenth Conference on Artificial intelligence/innovative Applications of Artificial intelligence. American Association for Artificial Intelligence, pp: 547-554. 1998.

[Coleman & Verma 1998]

Coleman, T.F. and Verma, A. *A preconditioned conjugate gradient approach to linear equality constrained minimization*. Technical report, Department of Computer Sciences, Cornell University, Ithaca, New York, USA, July 1998

[Costeira & Kanade 1998]

Costeira, J. and Kanade, T. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*. Vol. 29, N° 3. Sept. 1998. Pages:159-179.

[Chen et al. 2002]

Chen, T.; Hsu, Y.J.; Liu, X.; Zhang, W. *Principle component analysis and its variants for biometrics*. Proc. of the International Conference on Image Processing 2002. Vol.1, Pages: I-61-I-64. 2002.

[Christy & Horaud 1999]

Christy, S. and Horaud, R. *Iterative pose computation from line correspondences*. *Computer Vision and Image Understanding*. Vol. 73, Issue. 1 pp. 137-144. 1999.

[Darrell & Pentland 1991]

Darrell, T. and Pentland, A. *Robust estimation of a multi-layered motion representation*. Proc. of the IEEE Workshop on Visual Motion 1991. Pages:173-178.

[DeCarlo & Metaxas 2000]

DeCarlo, D. and Metaxas, D. *Optical Flow constraints on deformable models with applications to face tracking*. International Journal of Computer Vision, Vol. 38. pp. 99–127. 2000.

[DeMenthon and Davis 1995]

DeMenthon, D. and Davis, L.S. *Model-Based Object Pose in 25 Lines of Code*. International Journal of Computer Vision. Vol. 15. pp. 123-141, 1995.

[Fan et al 2008]

Fan W.; Zhenshen Q.; Changhong W.; Bin H. *Precise localization of indoor mobile robots in FMS based on distributed vision*. IEEE International Conference on Automation and Logistics, 2008. ICAL 2008. pp.2412-2416. 2008.

[Feng et al. 2009]

Feng W.; Kui Y.; Wei Z.; Xiaojie C.; Rui Z. *Visual navigation of an indoor mobile robot based on a novel artificial landmark system*. International Conference on Mechatronics and Automation, 2009. ICMA 2009. pp.3775-3780. 2009

[Fernandez et al. 2007]

Fernandez, I.; Mazo, M; Lázaro, J.L.; Pizarro, D.; Santiso, E; Martín, P.; Losada, C. *Guidance of a mobile robot using an array of static cameras located in the environment*. Autonomous Robots. Vol. 23, Issue 4, pp. 305-324. 2007.

[Finlayson et al. 2002]

Finlayson, G.D.; Hordley, S.D. and Drew, M.S. *Removing shadows from images*. Proc. of the 7th European Conference on Computer Vision 2002. Part IV. Pages: 823-836. 2002.

[Finlayson et al. 2004]

Finlayson, G.D.; Drew, M.S. and Lu, C. *Intrinsic Images by Entropy Minimization*. Proc. of the 8th European Conf. on Computer Vision 2004. Pages: 582-595. 2004.

[Finlayson et al. 2006]

Finlayson, G.D.; Hordley, S.D.; Lu, C. and Drew, M.S. *On the removal of shadows from images*. IEEE Pattern Analysis and Machine Intelligence (PAMI) Jan 2006. Vol.28, N°1, Pages: 59-68. 2006.

[Fu & Yang 2009]

Fu, S. and Yang, G. *Uncalibrated monocular based simultaneous localization and mapping for indoor autonomous mobile robot navigation*. International Conference on Networking, Sensing and Control, 2009. ICNSC '09. pp.663-668, 26-29. 2009.

[Gatica-Perez et al. 2007]

Gatica-Perez, D.; Lathoud, G.; Odobez, J.-M.; McCowan, I. *Audiovisual Probabilistic Tracking of Multiple Speakers in Meetings*. IEEE Transactions on Audio, Speech, and Language Processing. Vol.15, no.2, pp.601-616. 2007.

[Geman & Reynolds 1992]

Geman, D.; Reynolds, G., *Constrained restoration and the recovery of discontinuities*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.14, no.3, pp.367-383, 1992

[German 2003]

German K.M. Cheung. Visual Hull Construction, Alignment and Refinement for Human Kinematic Modeling, Motion Tracking and Rendering. Phd. Thesis. Robotics Institute Carnegie Mellon University. October 2003

[Haag & Nagel 1999]

Haag, M. and Nagel, H. Combination of edge element and optical flow estimates for 3-D model-based vehicle tracking in traffic image sequences. International Journal of Computer Vision. Vol. 35. pp. 295-319. 1999.

[Hartley & Vidal 2004]

Hartley, R.; Vidal, R. *The multibody trifocal tensor: motion segmentation from 3 perspective views*. Proc. of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., vol.1, pp. I-769-I-775, 2004.

[Hoff et al. 1996]

Hoff, W., Nguyen, K., and Lyon, T. *Computer vision-based registration techniques for augmented reality*. Proc. of Intelligent Robots and Computer Vision XV. SPIE . Vol. 2904. pp. 538-548. 1996.

[Hoover & Olsen 1999a]

Hoover, A.; Olsen, B.D. *A real-time occupancy map from multiple video streams*. Proc. of the 1999 IEEE International Conference on Robotics and Automation, vol.3, pp.2261-2266, 1999.

[Hoover & Olsen 1999b]

Hoover, A.; Olsen, B.D. *Path planning for mobile robots using a video camera network*. Proc. of the International Conference on Advanced Intelligent Mechatronics IEEE/ASME, 1999. pp.890-895, 1999.

[Hoover & Olsen 2000]

Hoover, A.; Olsen, B.D., *Sensor network perception for mobile robotics*. Proc. of the IEEE International Conference on Robotics and Automation, 2000. ICRA '00. vol.1, pp.342-347, 2000

[Horn & Schunck 1981]

Horn, B.K.P. and Schunck, B.G. *Determining Optical Flow*. Artificial Intelligence, N° 17. Pp:185-203. 1981.

[Hsieh et al. 2009]

Hsieh, H.; Wu, C.; Hu, J. A procedure for simultaneous calibration of environmental cameras and localization of mobile robots. 7th Asian Control Conference 2009. ASCC 2009.pp.905-912. 2009.

[Huabin et al. 2004]

Huabin, T.; Lei, W.; Zengqi, S. *Accurate and stable vision in robot soccer*. 8th Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004. Vol.3. pp. 2314-2319. 2004.

[Hung & Gatica-Perez 2008]

Hung, H.; Gatica-Perez, D. *Identifying dominant people in meetings from audio-visual sensors*. 8th IEEE International Conference on Automatic Face & Gesture Recognition, 2008. FG '08. pp.1-6. 2008.

[Hung et al. 2008]

Hung, H.; Huang, Y.; Yeo, C.; Gatica-Perez, D. *Associating audio-visual activity cues in a dominance estimation framework*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. pp.1-6. 2008.

[Jieping et al. 2004]

Jieping, Y.; Ravi, H.; Qi, L. *GPCA: an efficient dimension reduction scheme for image compression and retrieval*. Proc. of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining 2004. pp: 354-363. 2004.

[Kanatani 2001]

Kanatani, K. *Motion Segmentation by Subspace Separation and Model Selection*. Proc. of the 8th IEEE International Conference on Computer Vision 2001. Vol. 2, pp: 586-591. 2001.

[Kanungo et al. 2002]

Kanungo, T.; Netanyahu, N.S.; Wu, A.Y. *An efficient k-means clustering algorithm: analysis and implementation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, n° 7, July 2002.

[Kato et al. 2000]

Kato, H.; Billinghamurst, M.; Poupyrev, I.; Imamoto, K.; Tachibana, K. *Virtual object manipulation on a table-top AR environment*. Proc. of the IEEE and ACM International Symposium on Augmented Reality, 2000. (ISAR 2000). pp.111-119, 2000.

[Ke & Kanade 2004]

Ke, Q.; Kanade, T. *Robust subspace clustering by combined use of kNND metric and SVD algorithm*. Proc. of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. vol.2, pp. II-592-II-599. 2004.

[Kin 1993]

Kim, Y.H. *Localization of a mobile robot using a laser range finder in a hierarchical navigation system*. Proc. of the IEEE Southeastcon '93. pp.4-7. 1993.

[Kiriy & Buehler 2002]

Kiriy, E. and Buehler, M. *Three-state Extended Kalman Filter for Mobile Robot Localization*. McGill University. Montreal, Canada, Tech. Rep. TR-CIM 05.06, April, 2002.

[Koller et al. 1993]

Koller, D., Daniilidis, K., and Nagel, H. *Model-based object tracking in monocular image sequences of road traffic scenes*. International Journal of Computer Vision. Vol. 10. Issue: 3. pp. 257–281.1993.

[Koller et al. 1997]

Koller, D., Klinker, G., Rose, E., Breen, D., Whitaker, R., and Tuceryan, M. *Real-time vision-based camera tracking for augmented reality applications*. Proc. of the ACM Symposium on Virtual Reality Software and Technology. VRST '97. pp. 87-94. 1997.



[Koller-Meier 2001]

Koller-Meier, E.B. and Ade, F. *Tracking multiple objects using a condensation algorithm*. Journal of Robotics and Autonomous Systems, Vol. 34, pp. 93-105, February 2001.

[Laurentini 1991]

Laurentini, A. *The Visual Hull: a new tool for contour-based image understanding*. Proc. of the 7th Scandinavian Conference on Image Processing. pp: 993-1002. 1991.

[Laurentini 1994]

Laurentini, A. *The Visual Hull Concept for Silhouette-Based Image Understanding*. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 16 , Issue 2, pp: 150-162. February 1994.

[Lee & Hashimoto 2002]

Lee, J.; Hashimoto, H. *Intelligent Space - concept and contents*. Advanced Robotics; 2002, Vol. 16 Issue 3, Pages: 265-280. 2002.

[Lee et al. 2001]

Lee, J; Ando, N.; Yakushi, T.; Nakajima, K.; Kagoshima, T.; Hashimoto, H. *Adaptive guidance for mobile robots in intelligent infrastructure*. Proc. of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. vol.1. pp.90-95. 2001.

[Liang et al. 2008]

Liang, X.; Tomizawa, T.; Do H.M.; Kim, Y.; Ohara, K.; Kim, B.K.; Tanikawa, T.; Ohba, K. *Multiple robots localization using large planar camera array for automated guided vehicle system*. International Conference on Information and Automation, 2008. ICIA 2008. pp.984-990. 2008.

[Look & Shrobe 2004]

Look, G. and Shrobe, H. 2004. *A plan-based mission control center for autonomous vehicles*. Proc. of the 9th international Conference on intelligent User interfaces . IUI '04. ACM, New York, NY, 277-279. 2004.

[Look & Shrobe 2007]

Look, G. and Shrobe, H. 2007. *Towards intelligent mapping applications: a study of elements found in cognitive maps*. Proc. of the 12th international Conference on intelligent User interfaces IUI '07. ACM, New York, NY, 309-312. 2007.

[Look et al. 2005]

Look, G., Kottahachchi, B., Laddaga, R., and Shrobe, H. *A location representation for generating descriptive walking directions*. Proc. of the 10th international Conference on intelligent User interfaces 2005. IUI '05. ACM, New York, NY, 122-129. 2005,

[Lowe 1992]

Lowe, D. *Robust model-based motion tracking through the integration of search and estimation*. International Journal of Computer Vision. Vol. 8. Issue. 2. pp. 113-122. 1992.

[Luo et al. 2008]

Luo, R.C.; Jian-Xian Li; Chen, C.-T. *Indoor localization using line based map for autonomous mobile robot*. IEEE Workshop on Advanced robotics and Its Social Impacts, 2008. ARSO 2008. pp.1-6. 2008.

[Lloyd 1982]

Lloyd, S. *Least squares quantization in PCM*. IEEE Transactions on Information Theory. Vol. 28, n° 2, pp. 129-137. 1982

[Maciejewski & Roberts 2001]

Maciejewski, A.A.; Roberts, R.G. *An example of principal component analysis applied to correlated images*. Proc. of the 33rd Southeastern Symposium on System Theory. pp: 269-273. 2001.

[Marron 2008]

Marron, M. *Seguimiento de múltiples objetos en entornos interiores muy poblados basado en la combinación de métodos probabilísticos y determinísticos*. PhD. Tesis. Universidad de Alcalá, 2008

[Martinez & Kak 2001]

Martinez, A.M.; Kak, A.C. *PCA versus LDA*. IEEE Transactions on Pattern Analysis and Machine Intelligence. Feb. 2001. Vol. 23, N° 2, pp: 228-233. 2001.

[Mitiche & Sekkati 2006]

Mitiche, A. and Sekkati, H. *Optical Flow 3D Segmentation and Interpretation: A Variational Method with Active Curve Evolution and Level Sets*. IEEE Transactions on Pattern Analysis and Machine Intelligence 2006. Vol. 28, N° 11, Pp: 1818-1829. 2006.

[Mitiche 1994]

Mitiche, A. *Computational Analysis of Visual Motion*. Plenum Press. 1994.

[Mumford & Shah 1989]

Mumford, D. & Shah, J. *Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems*. *Communications on Pure and Applied Math.* Vol. 42, no. 5, pp: 577-684. 1989.

[Negerborn 2003]

Negenborn, R. *Robot Localization and Kalman Filters*. MS. Thesis. Institute of Information and Computing Sciences, Utrecht University. 2003

[Niitsuma et al. 2009]

Niitsuma, M.; Yokoi, K.; Hashimoto, H., *Describing human-object interaction in Intelligent Space*, 2nd Conference on Human System Interactions, 2009. HSI '09. pp.395-399. 2009

[Pentland 1997]

Pentland, A. *Smart rooms, desks and clothes*. 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997. ICASSP-97. Vol.1, pp.171-174. 1997.

[Pizarro et al. 2009]

Pizarro, D.; Mazo, M.; Santiso, E.; Marron, M.; and Fernandez, I. *Localization and Geometric Reconstruction of Mobile Robots Using a Camera Ring*. *IEEE Transactions on Instrumentation and Measurement*. Vol 58, No. 8. pp.2396-2409. 2009.

[Potamianos et al. 2006]

Potamianos, G.; Lucey, P. *Audio-Visual ASR from Multiple Views inside Smart Rooms*. 2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. pp.35-40. 2006.

[Rozgic et al. 2007]

Rozgic, V.; Busso, C.; Georgiou, P.G.; Narayanan, S. *Multimodal Meeting Monitoring: Improvements on Speaker Tracking and Segmentation through a Modified Mixture Particle Filter*. *IEEE 9th Workshop on Multimedia Signal Processing*, 2007. MMSP 2007. pp.60-65. 2007.

[Sekkati & Mitiche 2006a]

Sekkati, H. and Mitiche, A. *Joint Optical Flow Estimation, Segmentation, and Interpretation with Level Sets*. Computer Vision and Image Understanding 2006. Vol. 103, Nº 2, Pages: 89-100. 2006.

[Sekkati & Mitiche 2006b]

Sekkati, H. and Mitiche, A. *Concurrent 3D Motion Segmentation and 3D Interpretation of Temporal Sequences of Monocular Images*. IEEE Transactions on Image Processing 2006. Vol. 15, Nº 3, Pages: 641-653. 2006.

[Sekkati & Mitiche 2007]

Sekkati, H. and Mitiche, A. *A Variational Method for the Recovery of Dense 3D Structure from Motion*. Journal of Robotics and Autonomous Systems 2007. Vol 55, Pages: 597-607. 2007,

[Shi & Malik 2000]

Shi, J. and Malik, J. *Normalized cuts and image segmentation*. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2000, Vol. 22, Issue 8. pp: 888-905. 2000,

[Sogo et al. 1999]

Sogo, T., Ishiguro, H., and Ishida, T. *Acquisition of qualitative spatial representation by visual observation*. Proc. of the 16th international Joint Conference on Artificial intelligence. Vol. 2. pp.1054-1060. 2007.

[Sooyong & Jae-Bok 2007]

Sooyong L.; Jae-Bok S. *Mobile robot localization using infrared light reflecting landmarks*. International Conference on Control, Automation and Systems, 2007. ICCAS '07. pp.674-677. 2007.

[Steinhaus et al. 1999]

Steinhaus, P. Ehrenmann, M. and Dillmann, R. *MEPHISTO A Modular and Extensible Path Planning System Using Observation*. Lecture notes in computer science. Computer Vision Systems. Pp. 361-375. 1999

[Steinhaus et al. 2004]

Steinhaus, P.; Walther, M.; Giesler, B.; Dillmann, R. *3D global and mobile sensor data fusion for mobile platform navigation*. Proc. of the 2004 IEEE International Conference on Robotics and Automation. ICRA '04. , vol.4, pp. 3325-3330, 2004.

[Stiefelhagen et al. 2008]

Stiefelhagen, R.; Bernardin, K.; Ekenel, H.K.; Voit, M., *Tracking identities and attention in smart environments - contributions and progress in the CHIL project*. 8th IEEE International Conference on Automatic Face & Gesture Recognition, 2008. FG '08. pp.1-8, 17-19. 2008

[Turk & Pentland 1991]

Turk, M., Pentland, A. *Eigenfaces for recognition*. Journal of Cognitive Neuroscience 1991. Vol.3, Pages: 72-85. 1991.

[Vidal & Hartley 2008]

Vidal, R.; Hartley, R. *Three-View Multibody Structure from Motion*. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 30, no.2, pp.214-227. 2008.

[Vidal & Sastry 2002]

Vidal, R. and Sastry, S. *Segmentation of Dynamic Scenes from Image Intensities*. IEEE Workshop on Vision and Motion Computing. pp: 44-49. 2002.

[Vidal et al. 2003]

Vidal, R.; Yi M.; Sastry, S. *Generalized principal component analysis (GPCA)*. Proc. of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.1, pp. I-621-I-628. 2003.

[Vidal et al. 2005]

Vidal, R.; Yi M.; Sastry, S., *Generalized principal component analysis (GPCA)*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.27, no.12, pp.1945-1959, 2005.

[Vidal et al. 2006]

Vidal, R.; Ma, Y.; Soatto, S.; Sastry, S. *Two-View Multibody Structure from Motion*. International Journal of Computer Vision. Vol. 68. Issue: 1. pp: 7-25. 2006.

[Wang & Adelson 1993]

Wang, J.Y.A.; Adelson, E.H. *Layered representation for motion analysis*. Proc. of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1993. CVPR '93., pp.361-366, 1993.

[Weiser 1993]

Weiser, M. *Some computer science issues in ubiquitous computing*. Communications ACM. Vol. 36. N° 7. pp: 77-84. 1993.

[Weiser 1999]

Weiser, M. *The computer for the twenty-first century*. Scientific American, Vol. 265. Issue. 3 pp. 94–104. 1991

[Weiss 1996]

Weiss, Y. *A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models*. Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition. 1996. pp: 321-326. 1996.

[Wilkin & Huang 2007]

Wilkin, G.; Huang, X. *K-Means Clustering Algorithms: Implementation and Comparison*. Second International Multi-Symposiums on Computer and Computational Sciences, 2007. IMSCCS 2007. pp:133-136. 2007.

[Yan & Tjahjadi 2004]

Yan, H.; Tjahjadi, T.; *Multiple motion segmentation through a highly robust estimator*. IEEE International Conference on Systems, Man and Cybernetics 2004. pp: 3082-3087. 2004.

[Yang & Xinhe 2006]

Yang G.; Xinhe X., *Color Landmark Design for Mobile Robot Localization*. IMACS Multiconference on Computational Engineering in Systems Applications. Vol.2, pp.1868-1874. 2006

[Yu & Wenfei 2007]

Yu Z.; Wenfei L. *Preliminary Research on Indoor Mobile Robot Localization using Laser-activated RFID*. IEEE International Conference on RFID, 2007. pp.78-85, 26-28. 2007.

[Zhang et al 2004]

Zhang, P.; Milios, E.E.; Gu, J. *Underwater Robot Localization using Artificial Visual Landmarks*. IEEE International Conference on Robotics and Biomimetics, 2004. ROBIO 2004. pp.705-710. 2004

[Zhang et al. 2006]

Zhang, Z.; Potamianos, G.; Chu, S.M.; Tu, J.; Huang, T.S. *Person Tracking in Smart Rooms using Dynamic Programming and Adaptive Subspace Learning*. 2006 IEEE International Conference on Multimedia and Expo. pp.2061-2064. 2006

[Zhang et al. 2007]

Zhang, Z.; Potamianos, G.; Senior, A; Huang, T. *Joint face and head tracking inside multi-camera smart rooms*. *Signal, Image and Video Processing*, Volume 1, Number 2, June 2007 , Pages: 163-178(16)

