Universidad de Alcalá
Escuela Politécnica Superior

Grado en Ingeniería en Electrónica y Automática Industrial

Trabajo Fin de Grado

Contribución a la detección y conteo de personas a partir de información de profundidad

Autor: Álvaro Fernández Rincón

Tutores: Cristina Losada Gutiérrez y Marta Marrón Romera

2016

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería en Electrónica y Automática Industrial

Trabajo Fin de Grado

Contribución a la detección y conteo de personas a partir de información de profundidad

Autor: Álvaro Fernández Rincón

Directores: Cristina Losada Gutiérrez y Marta Marrón Romera

Tribunal:

Presidente: Carlos Andrés Luna Vázquez

Vocal 1º: José Luis Lázaro Galilea

Vocal 2º: Cristina Losada Gutiérrez

Calificación: .....

Fecha: .....

Dedicado a las personas que han estado, están y estarán apoyándome paso a paso en mi camino.

"El destino baraja las cartas pero nosotros somos quienes las jugamos"
William Shakespeare

Resumen

El objetivo de este trabajo es la detección robusta y el seguimiento de personas para su conteo. Para ello se parte de imágenes de profundidad (2.5D) adquiridas utilizando una cámara Kinect II ubicada en posición cenital. Esas imágenes se procesan para extraer descriptores que posteriormente se clasifican para determinar si se ha detectado una persona u otro elemento. Para la consecución del objetivo se han estudiado y evaluado diferentes descriptores. Además, se ha incluido un filtro de partículas extendido con proceso de clasificación (XPFCP) para el seguimiento de múltiples personas. El sistema se ha evaluado empleando una base de datos de imágenes de profundidad etiquetadas de forma manual, obteniéndose tasas de acierto medias del 97.7 %.

Palabras clave: Regiones de interés, XPFCP, conteo de personas, seguimiento de personas.

Abstract

The aim of this work is to develop a robust system for people detection, tracking and counting from depth images acquired using an overhead Kinect II camera. The images are processed in order to extract feature descriptors, and them a classificator is used in order to discriminate between people and other elements in the scene. To achieve this goal, two different descriptors have been analyzed and evaluated. Moreover, an extended particle filter with classification process (XPFCP) have been included for multiple people tracking. Several experimental tests have been carried out in order to validate the algorithm. A dataset manually labeled has been used, obtaining successful results with a true positive rate of 97.7 %.

Keywords: Region of interes, XPFCP, people counting, tracking.

Índice general

Resumen vii

Abstract ix

Índice general xi

Índice de figuras xv

Índice de tablas xvii

Índice de algoritmos xix

1 Introducción 1

1.1 Introducción al trabajo final de grado . . . 1

1.1.1 Objetivos . . . 1

1.1.2 Estructura del documento . . . 3

2 Fundamentos teóricos 5

2.1 Introducción . . . 5

2.2 Obtención de imágenes del entorno . . . 5

2.3 Obtención de información en profundidad mediante cámaras de tiempo de vuelo . . . 6

2.4 Kinect II . . . 8

2.5 Descriptores . . . 10

2.5.1 Descriptores de profundidad para reconocimiento de objetos . . . 12

2.5.1.1 Depth kernel descriptor . . . 12

2.5.1.2 Tamaño de la ventana de pixeles . . . 13

2.5.1.3 Forma de los objetos evaluados . . . 13

2.5.1.4 Obtención de información de los bordes . . . 14

2.5.1.5 Obtención de *match kernel* de forma piramidal . . . 15

2.5.2 Descriptores de profundidad para la detección de personas en imágenes cenitales . . . 16

2.5.2.1 Componentes primera, segunda y tercera . . . 17

2.5.2.2 Componentes cuarta y quinta . . . 18

2.5.2.3 Normalización de las cinco primeras componentes . . . 18
2.5.2.4 Componente sexta . . . 20
2.6 Clasificadores . . . 20
2.6.1 Maquina de soporte vectorial . . . 21
2.6.1.1 Clasificadores SVM con separación lineal . . . 21
2.6.1.2 Clasificadores SVM con separación cuasi-lineal . . . 24
2.6.1.3 Clasificadores SVM con separación no lineal . . . 26
2.6.2 Analisis de las componentes principales . . . 27
2.6.2.1 Metodo basado en correlaciones . . . 28
2.6.2.2 Metodo basado en covarianzas . . . 29
2.7 Sistemas de seguimiento . . . 30
2.7.1 Filtro de Kalman . . . 30
2.7.1.1 Predicción . . . 31
2.7.1.2 Corrección . . . 32
2.7.2 Filtro de partículas extendido . . . 32
2.7.2.1 Reinicialización . . . 33
2.7.2.2 Predicción . . . 34
2.7.2.3 Corrección . . . 34
2.7.2.4 Selección . . . 35
2.7.2.5 Modelo del sistema . . . 35
2.8 Conclusiones . . . 36

3 Implementación del sistema de detección y conteo de personas basado en imágenes de profundidad 37

3.1 Introducción . . . 37
3.2 Sistema de adquisición de imágenes . . . 38
3.3 Pre-procesado de las imágenes de entrada . . . 38
3.4 Obtención de la región de interés . . . 40
3.5 Extracción de características . . . 42
3.5.1 Descriptores de profundidad para reconocimiento de objetos . . . 42
3.5.1.1 Reducción de la dimensionalidad del vector de características . . . 47
3.5.2 Descriptores de profundidad para la detección de personas en imágenes cenitales . . . 49
3.6 Sistema de clasificación . . . 50
3.7 Sistema de seguimiento . . . 51
3.7.1 Etapas del Filtro de Partículas Extendido implementado . . . 51

4 Resultados **57**

4.1 Introducción . . . 57

4.2 Entorno experimental y base de datos utilizada . . . 57

4.3 Resultados obtenidos en función de los parámetros . . . 57

4.3.1 Métricas de calidad . . . 57

4.3.2 Resultados en función del porcentaje de partículas mantenidas entre iteraciones consecutivas . . . 58

4.3.3 Resultados en función de la distancia de asociación entre clusters . . . 59

4.3.4 Resultados en función del número de partículas del filtro de partículas extendido . 61

4.4 Comparación de resultados experimentales . . . 62

4.5 Conclusiones . . . 63

5 Conclusiones y líneas futuras **65**

5.1 Conclusiones . . . 65

5.2 Líneas futuras . . . 66

Bibliografía **67**

A Manual de usuario **69**

A.1 Introducción . . . 69

A.2 Manual . . . 69

A.3 Ejecución del software . . . 71

B Herramientas y recursos **73**

C Pliego de condiciones **75**

C.1 Requisitos de Hardware . . . 75

C.2 Requisitos de Software . . . 75

D Presupuesto **77**

D.1 Costes de equipamiento . . . 77

D.2 Costes de mano de obra . . . 77

D.3 Coste total del presupuesto . . . 78

Índice de figuras

1.1 Diagrama de las 2 fases principales del proceso de detección y conteo de personas. 2

2.1 Imagen de la posición de la cámara, cenital, dentro de entorno de estudio. 6

2.2 Imagen de la emisión y recepción de la onda enviada para la adquisición de imágenes con las cámaras de tiempo de vuelo. 7

2.3 Funcionamiento de los transistores MOSFET durante los 4 períodos de recepción de el haz de luz, cuyo estudio se detalla en [1] de donde procede esta figura. 7

2.4 Esquema de los diferentes bloques que forman el sistema de adquisición de la kinect II, diagrama extraído de [2] 9

2.5 Ejemplos de los valores de los momentos invariantes de *Hu* para diferentes caracteres. . . 12

2.6 Esquema de las diferentes subregiones de idéntica altura que fragmentan la región evaluada. 17

2.7 Esquema de las diferentes subregiones de idéntica altura que fragmentan la región evaluada. 19

2.8 Gráfica de $\rho_1^{ROI_{r,c}^k}$, donde se muestra el conjunto de datos de entrenamiento y la curva ajustada, así como los valores del conjunto de coeficientes a_0 , a_1 y a_2 y el error cuadrático medio obtenido. 19

2.9 (a)Ejemplo de hiperplano de separación. (b) Diferentes ejemplos de hiperplanos de separación que muestran las infinitas posibilidades. (c) Ejemplo de hiperplano no óptimo con margen no máximo. (d) Ejemplo de hiperplano óptimo con margen de separación máximo 22

2.10 Ejemplo de las muestras no-separables, de su variables de holgura, las cuales indican la desviación desde el borde del margen de la clase respectiva. Los ejemplos $x_i, x_j y x_k$ son no-separables ya que sus variables de holgura son mayores que cero, pero mientras que $x_j y x_k$ están mal clasificados al estar en lado incorrecto de la frontera de decisión, x_i esta bien clasificado 25

2.11 Ejemplo de transformación de un espacio de entrada inicial hasta el espacio de características, en el cual se busca la decisión lineal 27

2.12 Diagrama de las 2 fases principales del proceso de detección y conteo de personas. 28

2.13 Diagrama de las etapas del Filtro de Kalman extraído de [3]. 31

2.14 Diagrama de las 2 fases principales del proceso de detección y conteo de personas. 33

3.1 Diagrama de las 2 fases principales del proceso de detección y conteo de personas. 37

3.2 Imagen obtenida de un *frame* del entorno evaluado sin tratamiento del ruido. 39

3.3 Imagen obtenida de un *frame* del entorno evaluado tras el tratamiento del ruido. 40

3.4 Niveles de vecindad y direcciones de búsqueda de las subregiones pertenecientes a la ROI. 41
3.5 Vectores de características de regiones de interés clasificadas como personas. 44
3.6 Vectores de características de regiones de interés clasificadas como no personas. 45
3.7 Varianza de las 14000 características de los vectores de características de regiones de interés clasificadas como personas. 46
3.8 Varianza de las 14000 características de los vectores de características de regiones de interés clasificadas como no personas. 46
3.9 Covarianza de las 14000 características de los vectores de características de regiones de interés evaludas. 47
3.10 Indice de correlación de las 14000 características de los vectores de características de regiones de interés evaluadas. 47
3.11 Ratio de Fisher de las 14000 características de los descriptores. 47
3.12 Autovalores de las componentes principales evaluadas. 48
3.13 Ejemplos de diferentes vectores de características, los vectores de color rojo representan a una persona de 180cm y pelo corto, mientras que los de color azul representan a una persona de 162cm y pelo largo. 49
3.14 Ejemplos del vector de 6 componentes obtenido a partir de el vector previo de 20 componentes, cuya dimension se reduce para aumentar la robustez. El vector a) corresponde a una persona de 165cm, b) a una de 183cm, c) a una de 187cm y d) a una de 197cm de altura. 49
3.15 Diagrama de bloques del algoritmo k-medias extendido empleado en la etapa de clasificación de las partículas del sistema. 53

4.1 Imagen de uno de los frames de la secuencia de empleada para el análisis paramétrico. . . 58
4.2 Análisis paramétrico en función de $N(\%)_{PARTIC_{NEW}}$ evaluando TP(%) y FN(%). 59
4.3 Análisis paramétrico en función de $Distancia_{MAX}$ evaluando TP(%) y FN(%). 60
4.4 Análisis paramétrico en función de N_{PARTIC} evaluando TP(%) y FN(%). 61
4.5 Secuencia de un frame con 5 personas del sistema con la etapa de seguimiento implementada. 63

A.1 Ejemplo del fichero de extensión *.result*. 71
A.2 Ejemplo del fichero de extensión *.xpf*. 72

Índice de tablas

3.1 Personas empleadas en el entrenamiento del clasificador PCA. 50

4.1 Resultados comparativos de la etapa de seguimiento sin la etapa de seguimiento. 62
4.2 Resultados comparativos de la etapa de seguimiento con la etapa de seguimiento. 62

D.1 Costes del equipamiento Hardware empleado. 77
D.2 Costes del equipamiento Software empleado. 77
D.3 Costes de la mano de obra empleada. 77
D.4 Coste total del presupuesto. 78

Índice de algoritmos

3.1 Algoritmo del código empleado para la asociación de las personas entre la entrada y salida del sistema de seguimiento 54
3.2 Algoritmo del código empleado para la consecución del histórico del sistema de seguimiento 55

Capítulo 1

Introducción

Justifica tus limitaciones y te quedaras con ellas.

Richard Bach

1.1 Introducción al trabajo final de grado

En la actualidad nos encontramos ante la necesidad de prevenir y alertar ante situaciones que pongan en riesgo a las personas, por ello la existencia de sistemas de control de aforo resulta cada día más importante. Pero cuando se tratan temas tan sensibles como la identificación de personas, puede verse afectado el derecho a intimidad de las mismas.

En los sistemas actuales se emplean cámaras de alta definición, pero ello implica un amplio grado de invasión de la intimidad de las personas, también se cuenta con sistemas de detección por barrera que ofrecen una seguridad y efectividad muy mejorable. El grupo de investigación GEINTRA (Grupo de Ingeniería Electrónica aplicada a Espacios Inteligentes y Transporte) cuenta con diversas líneas de investigación, entre las que se encuentra la detección y conteo de personas donde se centra este Trabajo Final de Grado.

En este contexto, en el presente trabajo se implementa un sistema de detección y conteo de personas empleando imágenes en profundidad, que son suministradas por una cámara de tiempo de vuelo. Estas cámaras ofrecen información de profundidad definida como la distancia de cada punto del entorno a la lente de la cámara. A partir de esta información es posible detectar a las personas que se encuentran dentro del entorno, pero con la ventaja de que no permiten identificar a las personas que son detectadas. Esto permite implementar un sistema de detección y conteo de personas no invasivo con la intimidad de las personas, pero a su vez robusto y fiable.

Una de las grandes ventajas de los sensores de profundidad es la posibilidad de emplearlos en entornos oscuros o con iluminación muy variable, ya que la imagen que se obtiene a su salida es invariante a la iluminación.

1.1.1 Objetivos

El objetivo de este trabajo final de grado consiste en realizar un sistema de detección y conteo de personas completo y robusto. Para lograrlo se divide en varias etapas el proceso, centrándose la primera etapa en la investigación de las diferentes características que se pueden extraer de las imágenes proporcionadas

por los sensores de profundidad. Una vez seleccionado el método mas robusto para el tipo de imágenes con las que se trabaja, así como para la posición cenital de la cámara de tiempo de vuelo, se centra el objetivo en la obtención de modelos de persona y no persona, para lo que se entrenan diversos modelos en la fase off-line. Una vez obtenidos tales modelos, en la fase on-line nos centramos en la adquisición, filtrado y definición de la región de interés (ROI), de la cual se extraen características para su posterior clasificación y determinación de su clase, persona o no persona. Finalmente se procede a realizar una fase de *tracking*, en la que se determina la continuidad de una o varias personas dentro del entorno durante una secuencia continuada, así como si orientación y velocidad. En el diagrama 3.1 se pueden observar las diferentes etapas anteriormente mencionadas.

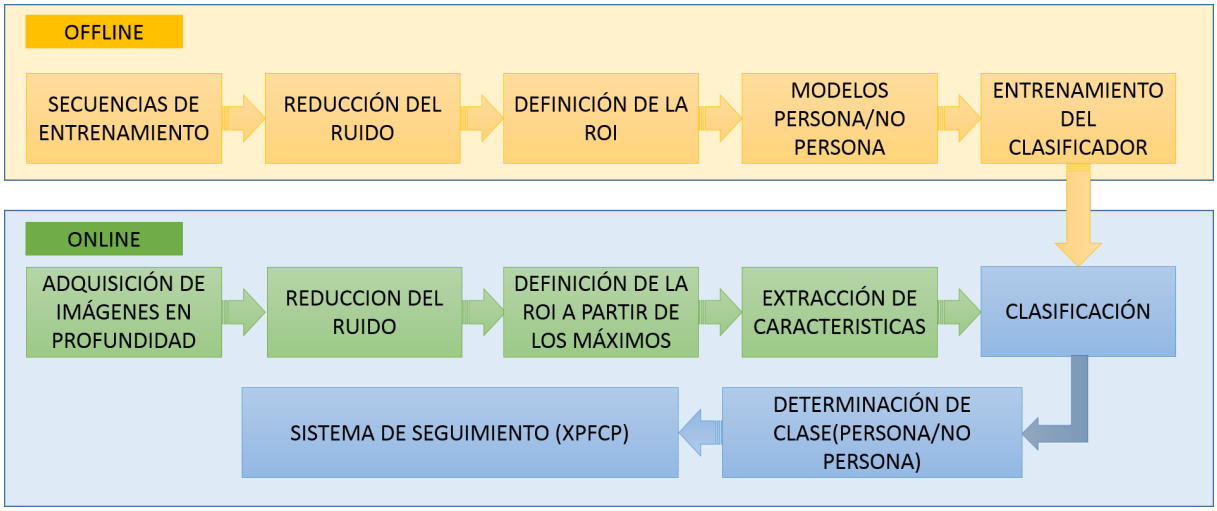

Figura 1.1: Diagrama de las 2 fases principales del proceso de detección y conteo de personas.

Como se muestra en el diagrama 3.1, para lograr el objetivo que se comentó anteriormente, son necesarias 2 etapas principales, que a su vez constan de diferentes pasos descritos a continuación:

Offline: *etapa en la cual se parte de secuencias de entrenamiento que representan ampliamente las posibles personas a evaluar, para obtener finalmente los modelos de persona y no persona, que permiten clasificar los candidatos*

- Evaluación de las secuencias de entrenamiento obtenidas mediante la cámara de tiempo de vuelo TOF. Se obtienen y procesan las imágenes del entorno para poder trabajar con ellas.

- Reducción de ruido: las imágenes proporcionadas por las cámaras de tiempo de vuelo contienen una cantidad de ruido considerable, por lo que se elimina el ruido mediante un proceso en el cual se asocia al valor de un píxel nulo, el valor de su píxel más cercano no nulo.

- Definición de la ROI a evaluar: dado que se trata de una fase de entrenamiento, se define la ROI según las necesidades, para ello se emplea un proceso en el cual se seleccionan 6 puntos característicos de la persona evaluada, 3 correspondientes a su cabeza y 3 de sus hombros y nuca. Conforme a tales puntos se selecciona la región de interés a evaluar.

- Estimación de los modelos que se evaluaran en la fase Online: tras la obtención de las regiones de interés se procede a la extracción de características, tanto de las ROI's de las cuales se conoce que pertenecen a personas, como de no personas. A partir de los vectores de características de estas dos clases, se forman modelos, los cuales determinarán el resultado de la clasificación en la etapa on-line

- Optimización de los modelos y obtención de un clasificador: con los vectores de características de los modelos obtenidos en la fase anterior, se entrena un clasificador, es decir se introducen un conjunto de muestras indicando su clase y se entrena el clasificador, para que al introducirle el vector de características de una imagen no contemplada en el entrenamiento, distinga la clase de pertenencia correctamente.

Online: *etapa en la cual partiendo de las secuencias adquiridas en tiempo real y de los modelos obtenidos en la fase offline, se procede a la clasificación de los candidatos del entorno evaluado, y tras ello a su seguimiento mediante el Filtro de Partículas Extendido con Proceso de Clasificación*

- Adquisición de imágenes de profundidad mediante la cámara de tiempo de vuelo TOF: se procede a adquirir y procesar las imágenes obtenidas por la cámara en posición cenital.
- Reducción del ruido de las imágenes: las imágenes proporcionadas por las cámaras de tiempo de vuelo contienen una cantidad de ruido considerable, por lo que se elimina el ruido mediante un proceso en el cual se asocia al valor de un píxel nulo, el valor de su píxel mas cercano no nulo.
- Definición de la ROI a partir de los máximos detectados: dado que en el proceso on-line no se puede seleccionar la región de interés de forma manual como en el caso off-line, se determina la región de interés en función de los máximos encontrados y la forma de la nube de puntos que se encuentran a su alrededor. Con lo que se obtiene la región o regiones de interés en un instante de tiempo determinado.
- Extracción de características de la ROI obtenida.: tras la obtención de la ROI se procede a la extracción de sus características con los descriptores seleccionados, tras o cual se contará con un vector de características que representa las características mas importantes de la ROI.
- Determinación de la existencia o no de una o múltiples personas en la imagen gracias al clasificador obtenido de la fase Off-line: una vez obtenido el vector de características de la ROI a evaluar se introduce en el clasificador, el cual indica la pertenencia a una de las clases con las que se realizó su entrenamiento. A la salida del entrenador ya se contará con la información necesaria para determinar la existencia de personas en la región de interés.
- Sistema de seguimiento: una vez conocida la existencia de una o varias personas dentro de la región de interés se procede a evaluar su posición, orientación y velocidad, para lo cual se realiza un sistema de *traking* basado en un *Filtro de Partículas Extendido*.

El sistema de detección y conteo que se especifica, se basa en una implementación en lenguaje C/C++ que hace uso de librerías de visión como OpenCV, aplicadas a las imágenes en profundidad aportadas por el sensor de infrarrojos de la Kinet II.

1.1.2 Estructura del documento

El trabajo final de grado que se trata en este documento se estructura en 5 secciones, las cuales se exponen a continuación, explicando brevemente su contenido:

1. Introducción al trabajo final de grado: donde se expone la necesidad de los sistemas de detección y conteo de personas, debido a las necesidades actuales para la seguridad de las personas. Para ello se exponen las ventajas de las cámaras TOF empleadas en este trabajo.

2. Fundamentos Teóricos: a lo largo de este capítulo se estudiarán las distintas teorías en las que se apoya este trabajo, así como las diversas opciones que se han estudiado hasta llegar a una solución fiable y robusta.

3. Desarrollo e implementación del sistema: en este capítulo se tratarán todos los bloques y detalles del sistema propuesto, tanto para la fase *online*, como *offline* mostradas en el diagrama 3.1.

4. Resultados experimentales: en este capítulo se evaluarán los resultados obtenidos tras la evaluación de unas imágenes obtenidas anteriormente que muestren diferentes situaciones.

5. Conclusiones y líneas futuras: a lo largo de este capítulo se evaluarán los resultados obtenidos dando lugar a las explicaciones correspondientes que argumentan el procedimiento elaborado.

Capítulo 2

Fundamentos teóricos

2.1 Introducción

Como ya se ha comentado en la introducción, el objetivo de este trabajo consiste en lograr la detección de personas dentro de un entorno de forma robusta, para ello se parte de información de profundidad proporcionada por una cámara Kinect II [4], la cual se encuentra situada en posición cenital, como se puede observar en la imagen 2.1 . Por tanto la única información con la que cuenta el sistema es de profundidad, la cual es procesada y transcurre por diversas etapas, hasta poder finalizar el proceso determinando si dentro del entorno se encuentran o no, una o varias personas.

El sistema implementado cuenta con diversas etapas, dentro de la cuales se han evaluado diferentes opciones hasta llegar a la solución óptima para nuestro objetivo. En este capítulo se exponen los fundamentos teóricos de los que se hace uso en este Trabajo Final de Grado.

2.2 Obtención de imágenes del entorno

En el sistema propuesto se comienza obteniendo imágenes del entorno, para ello es necesario determinar tanto el tipo de tecnología que se emplea para obtenerla, como la posición de la cámara, ya que de ello dependerá todo el proceso. Por ello partimos de unos requisitos que nos permitirán obtener un sistema que no viole la intimidad de las personas, así como fiable y robusto.

La primera cuestión que se plantea es la tecnología que se usará, para ello se estudian las diferentes cámaras que encontramos en el mercado en la actualidad. Evaluando el mercado, se observa la posibilidad de obtener imágenes en color (RGB), escala de grises o imágenes en profundidad.

Las imágenes en color y en escala de grises no cumplen uno de los requisitos que se imponen a nuestro sistema, ya que aportan información que podría servir para identificar al individuo que se evalúa, dando lugar a una violación de su intimidad y la necesidad del almacenaje de la información obtenida así como de las consecuente legalización de la situación de las mismas. Por ello, se emplean en nuestro sistema imágenes de profundidad, las cuales proporcionan información de la cual se pueden extraer características que nos permiten detectar personas dentro del entorno, pero que no dan lugar por su calidad a identificar a la persona en sí.

En nuestro caso queremos identificar a una o varias personas, por lo que la situación en la que colocamos la cámara juega un papel fundamental, por ello debemos de evitar oclusiones entre los elementos

que se encuentran en el entorno. En vista de ello la cámara TOF será situada en posición cenital tal y como se muestra en la imagen 2.1.

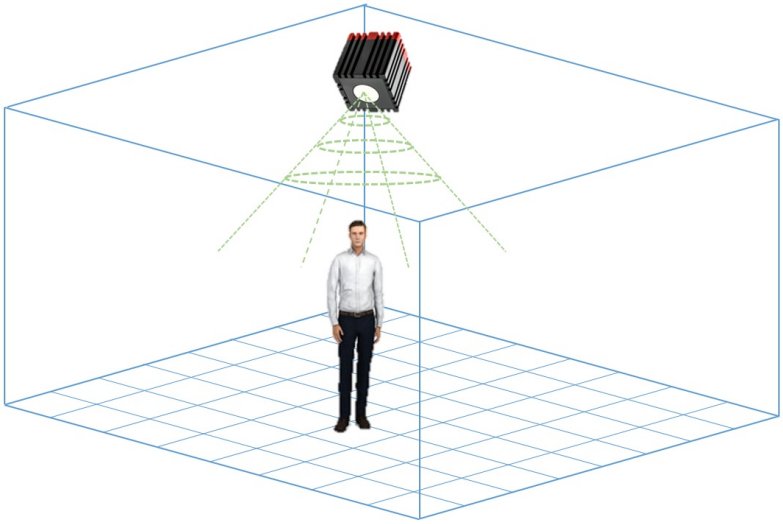

Figura 2.1: Imagen de la posición de la cámara, cenital, dentro de entorno de estudio.

2.3 Obtención de información en profundidad mediante cámaras de tiempo de vuelo

A continuación se muestra el principio de funcionamiento de las cámaras de tiempo de vuelo (TOF) .

Las cámaras con sensores convencionales constan de matrices de celdas, las cuales almacenan la luz dando lugar a un valor digital. Por lo cual cada celda tiene carácter monocromático, es decir su valor representa un único color, en el caso de los sensores RGB cada píxel se forma por la evaluación de 4 celdas contiguas. La intensidad de cada celda dependerá de los fotones que incidan sobre cada una de las mismas, dando lugar todas ellas a los píxeles que formarán la imagen final.

En este sistema se adquieren imágenes con la técnica de tiempo de vuelo, la cual proporciona información de profundidad o 2.5D sin el habitual uso de sistemas estereoscópicos. Los sistemas TOF envían haces de luz infrarroja modulada a una determinada frecuencia hacia la escena, en el caso de que el haz de luz se encuentre con algún ítem en su recorrido, rebotará regresando hasta un receptor de la cámara. Dado que el haz de luz se envía modulado con una determinada frecuencia, la diferencia de fase entre el haz enviado y el recibido proporciona la distancia a la que se encuentran los diferentes objetos de la escena de forma indirecta. El sistema cuenta con dos elementos principales, el emisor y el receptor.

- **Emisor:** en la actualidad el sistema emisor más empleado son los anillos de luz infrarroja. Los haces de luz que envía están modulados a una frecuencia determinada, habitualmente cercana al infrarrojo, 850nm, por lo que no son perceptibles por el ojo humano. Esta característica es una gran ventaja para nuestro sistema, ya que la situación lumínica del entorno en el que se implante el sistema no influirá excesivamente en las imágenes obtenidas. Con ello evitaremos los habituales problemas de iluminación que se suelen encontrar en visión.

  La modulación de onda continua es la más empleada, debido a que el espacio es iluminado de forma continuada con los haces de luz modulados. Gracias a la diferencia de fase entre el haz de luz enviado y el recibido se obtiene la distancia, tal y como se muestra en el esquema de la imagen 2.2.

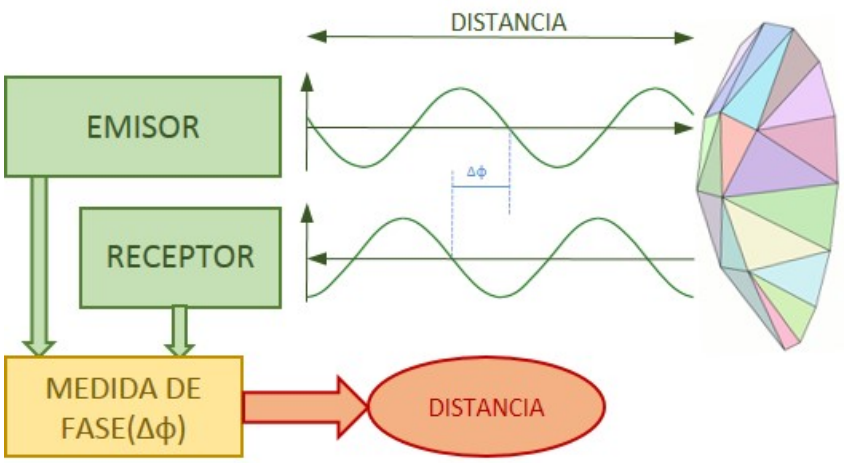

Figura 2.2: Imagen de la emisión y recepción de la onda enviada para la adquisición de imágenes con las cámaras de tiempo de vuelo.

- **Receptor:** el haz de luz rebotado tiene una diferencia de fase al analizarlo frente al enviado, gracias a lo cual podemos obtener la distancia a la que se encuentra el objeto. El receptor no solo recibe componentes de haz rebotados sino también componentes del ambiente que producen ligeros errores en la medida.

Actualmente el funcionamiento de las cámaras TOF esta basado en la emisión y recepción de haces de luz con una frecuencia determinada. Como se comenta en [5] para la obtención de la distancia a la que se encuentra el elemento estudiado se emite un haz de luz a una frecuencia conocida, f_{mod}, el haz de luz rebota en el ítem que se esta estudiando, regresando al sensor con un desplazamiento conforme al haz enviado, el análisis del desplazamiento durante un periodo T permite obtener la distancia a la que se encuentra. Para ello se divide el periodo en 4 desplazamientos de 90º, evaluando la diferencia de fase entre los dos haces de luz, emitido y rebotado. El sistema cuenta con receptores formados por transistores MOSFET cuya carga acumulada en cada uno de los 4 períodos indica la diferencia de fase, como muestra la imagen 2.3.

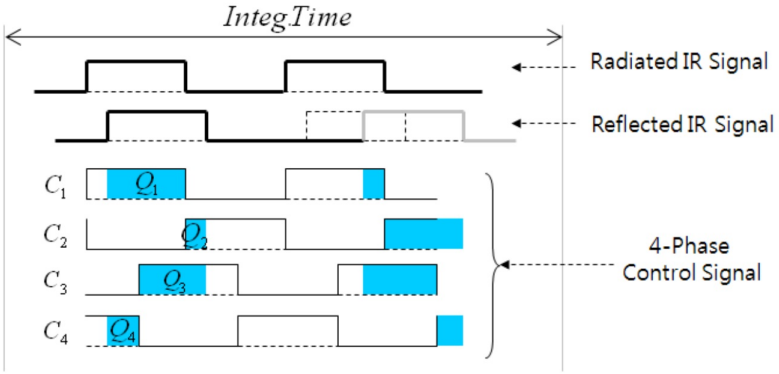

Figura 2.3: Funcionamiento de los transistores MOSFET durante los 4 períodos de recepción de el haz de luz, cuyo estudio se detalla en [1] de donde procede esta figura.

En la imagen 2.3 se indica el principio de evaluación de la diferencia de fase entre el haz recibido y el enviado, dando lugar a las cargas acumuladas en los 4 periodos de 90º.

En la ecuación se calcula la diferencia de fase entre el haz emitido y el reflejado, β , a partir de las cargas acumuladas Q_A, Q_B, Q_C y Q_D.

$$\beta = arctan(\frac{Q_C - Q_D}{Q_A - Q_B}) \tag{2.1}$$

Una vez conocida la diferencia de fase se puede obtener la distancia a la que se encuentra el ítem, d, debida la dependencia que tiene de la velocidad de la luz en el vacío $C = 3x10^8 m/s$ y la frecuencia de modulación del rayo emitido f_{mod}.

$$d = \frac{c}{4\pi f_{mod}}\beta \tag{2.2}$$

Partiendo de la formula anterior se obtiene la distancia máxima a la que se pueden medir objetos de forma fiable, d_{MAX}. La diferencia de fase entre los haces de luz debe encontrarse entre 0 y 2π radianes. Si se intenta obtener la distancia a objetos más lejanos que d_{MAX}, se obtendrán medidas erróneas. La máxima diferencia de fase que se puede medir sin ambigüedad es de 2π radianes, ya que a partir de este valor la señal se repetiría, sin poder conocer si la diferencia de fase entre los haces de luz es de β o $\beta + 2\pi$.

$$d_{MAX} = \frac{c}{2f_{mod}} \tag{2.3}$$

De la fórmula anterior se observa que la distancia máxima mesurable depende inversamente de la frecuencia moduladora del haz de luz emitido por el anillo de infrarrojos.

En la sección que se tratará a continuación se estudiaran las características de la cámara seleccionada.

2.4 Kinect II

La cámara Kinect II [2] ha sido desarrollada por Microsoft para su uso en la *XBOX ONE*, así como en situaciones comunes. Se trata de una cámara que aporta a su salida imágenes en 3 formatos diferentes: RGB, escala de grises e imágenes de profundidad. La cámara Kinect II proporciona imágenes de buena calidad a un precio considerablemente más económico que las que se suelen encontrar en el mercado, por ello ha sido seleccionada para este trabajo. En concreto, se emplean imágenes de profundidad que contienen ruido, por lo que debe ser reducido en etapas posteriores del sistema. La cámara Kinect II ha evolucionado notablemente con su anterior comercialización, y en la actualidad ofrece las siguientes características:

- Ángulo de visión (FOV) 70º horizontalmente y 60º verticalmente.
- Medidas válidas comprendidas entre 0.8m y 4.2m.
- Imágenes de profundidad de 512x424 píxeles.
- Resolución de profundidad dentro del 1 % de la distancia medida.
- Apertura focal F=1.1.
- Tiempo de exposición máximo de 14ms.
- Menos de 20 ms de latencia al principio de cada exposición a los datos durante la entrega a través del USB 3.0 al sistema principal.
- Error en la medida menor de 2 % dentro del rango de operaciones.

En la imagen 2.4 se puede apreciar un esquema de los bloques que forman el sistema de la *Kinect II*, donde se observa el generador de los haces modulados emitidos, las etapas de emisión y recepción de los haces así como su posterior etapa de adquisición y adecuación en el *SoC* (Sistem on a Chip).

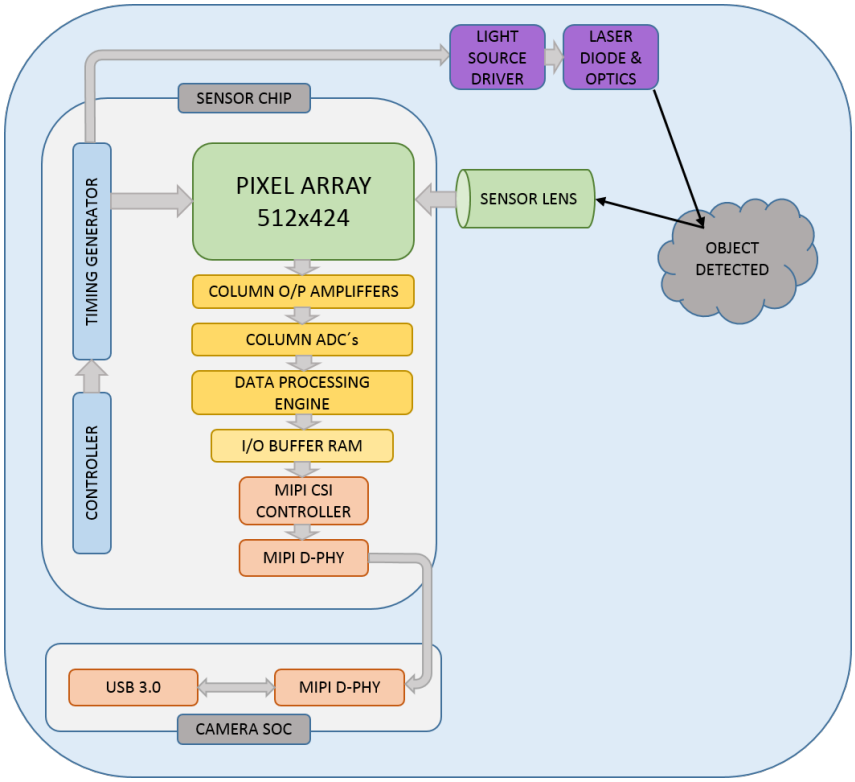

Figura 2.4: Esquema de los diferentes bloques que forman el sistema de adquisición de la kinect II, diagrama extraído de [2]

La cámara empleada, la Kinect II, contiene las mismas fuentes de error que las cámaras TOF mas comunes, se trata de los errores sistemáticos y los no sistemáticos. Tales errores afectan a las medidas de profundidad realizadas, viéndose afectada la precision y fiabilidad de la cámara. Las características de los errores se muestran a continuación.

- **Errores sistemáticos:** estos errores suelen aparecer a la hora de realizar la transformación de la luz recibida en una señal adecuada, y se suelen compensar mediante correcciones hardware. Entre los errores mas comunes destacan:
  - Error Wiggling: se trata de la fuente de error mas crucial, se produce al alterar la distancia entre la cámara y la superficie en medidas, (principalmente por el vaivén de la cámara en su sujeción), lo que altera la medida obtenida. Para evitarlo se emplean funciones de correlación que incorporan modos de Fourier superiores. Por ejemplo mediante el uso de la superposición de finitas ondas coseno superpuestas.
  - Ruidos de patrón fijo en diversos píxeles.
  - Variación en la amplitud: se produce debido a la iluminación y la reflectividad, ya que en un medio real, aun estando controlado nunca se mantiene constante de manera perfecta.
  - Ruido de disparo (shotnoise): se trata de un ruido electrónico debido al bajo numero de electrones en los transistores Mosfet, ya que pequeñas fluctuaciones en ellos producen variaciones en las mediciones.

  - Errores debido a la temperatura: el material semiconductor de los transistores puede verse afectado por la variación de la temperatura del entorno en el que s encuentra el sensor.

- **Errores no sistemáticos:** se tratan de aquellos errores no predecibles que se dan en las mediciones.

  - *Flying* píxeles: se deben a discontinuidades en los items que se encuentran en el entorno evaluado.
  - Medición de la distancia máxima mesurable: como se comentó anteriormente la diferencia de fase entre los haces de luz debe encontrarse entre 0 y 2π radianes. Si se intenta obtener la distancia a objetos más lejanos que d_{MAX}, se obtendrán medidas erróneas.
  - Artefactos de movimiento: debidos a la existencia de movimiento durante el tiempo de integración que provoca que para un mismo punto se obtengan dos distancias diferentes.
  - Interferencias entre las señales que se reciben por el sensor debido al multicamino.

2.5 Descriptores

Los descriptores son las diferentes formas de extraer características de una imagen para su posterior procesamiento. Con ello se puede obtener información objetiva de la imagen que permite distinguir los elementos de la escena. En la actualidad existen multitud de descriptores para la extracción de características. A continuación se presentan los más utilizados:

Histogramas: esta técnica consiste en evaluar los atributos individuales de cada uno de lo píxeles que forman la imagen, obteniendo los valores de intensidad de los colores que lo forman, tres en las imágenes RGB y uno en las imágenes de profundidad y las de escala de grises. Con ello se obtiene información del grado de repetición de los valores de los píxeles de la escena. En ciertos casos esta técnica puede ser suficiente para discriminar entre los objetos o clases que se evalúan. Pero en nuestro caso dado la complejidad de las imágenes que se evaluan no resulta una técnica acertada, por lo que es necesario analizar otras opciones.

Descriptores basados en perímetro, área, circularidad, rectangularidad y momentos:

- Los descriptores basados en perímetro y área: son descriptores muy básicos, solo pueden ser aplicados en imágenes binarias por lo que se reduce en gran medida su área de aplicación. El perímetro consiste en calcular los píxeles que pertenecen al objeto evaluado y que tienen al menos un píxel vecino perteneciente al fondo umbral utilizado. Mientras que el área consiste en el cálculo de los píxeles que pertenecen al objeto en sí.

  - El área de una región, R , viene descrita como el número de puntos que se encuentran dentro de su contorno.

$$Area = \sum píxeles \in R \tag{2.4}$$

  - El perímetro de una región, R, se obtiene empleando un código cadena como la suma del número de desplazamientos horizontales, N_H, verticales, N_V, y diagonales, N_D, estando estos últimos multiplicados por un factor $\sqrt{2}$.

$$Perimetro = N_H + N_V + N_D\sqrt{2} \tag{2.5}$$

- Descriptores basados en circularidad y rectangularidad: gracias a estos momentos se puede obtener una medida aproximada de la forma que tiene el objeto, en el caso de ser un objeto cuya proyección

se asimila a una circunferencia, la circularidad será cercana a 1 como se puede observar en las ecuaciones que se muestran a continuación. La rectangularidad el grado en el que el objeto evaluado se asemeja a un rectángulo incluyendo el menor fondo posible.

$$Circularidad = 4\pi \frac{Area}{Perimetro^2} \tag{2.6}$$

$$Rectangularidad = \frac{Ancho \times Alto}{Area} \tag{2.7}$$

- Descriptores basados en los momentos: el objeto evaluado tiene momentos que lo caracterizan frente a los demás objetos. Se pueden obtener momentos invariantes a la posición del objeto dentro de la escena, momentos invariantes a la posición y escalado así como momentos invariantes a la posición, escalado y rotación. A continuación se muestra su cálculo.

  Para el calculo de los momentos se comienza por obtener el centroide, el cual viene descrito por las coordenadas, $\bar{x}$ e $\bar{y}$.

$$\bar{x} = \frac{1}{|R|} \sum_{(u,v)\in R} u \tag{2.8}$$

$$\bar{y} = \frac{1}{|R|} \sum_{(u,v)\in R} v \tag{2.9}$$

  A partir del centroide se calculan los diferentes momentos, tanto momentos de orden (p, q) como momentos centrales, μ_{pq}, centrales normalizados, η_{pq}, o momentos invariantes de Hu, [6].

$$\mu_{pq} = \sum_{(u,v)\in R} (u - \bar{x})^p (v - \bar{y})^q \tag{2.10}$$

$$\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{(\frac{p+q}{2}+1)}} \tag{2.11}$$

  Una vez obtenidos los momentos invariantes a la posición y escalado, se pueden calcular los momentos que son a su vez invariantes a la rotación, conocidos como momentos invariates de Hu, η.

$$h_1 = \eta_{20} + \eta_{02} \tag{2.12}$$

$$h_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{2.13}$$

$$h_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{2.14}$$

$$h_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{2.15}$$

$$h_5 = (\eta_{30}-3\eta_{12})(\eta_{30}+\eta_{12})((\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2)+(3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})(3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2) \tag{2.16}$$

$$h_6 = (\eta_{20} - \eta_{02})((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \tag{2.17}$$

$$h_7 = (3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})(3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2)-(\eta_{30}-3\eta_{12})(\eta_{21}+\eta_{03})(3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2) \tag{2.18}$$

Los descriptores que emplean los momentos invariantes de Hu se suelen emplear en los sistemas de cintas automatizadas para la detección de piezas erróneas. Ello se debe a que suelen tener el mismo perfil y siempre circulan por la misma sección, y dado que se trata de descriptores invariantes a la traslación y

rotación se obtienen resultados muy buenos. También se suele emplear para la identificación de caracteres, letras o números, debido a que su representación suele ser de una forma concreta y constante, y al igual que anteriormente se obtienen buenos resultados debido a la invarianza a la traslación y rotación. En la imagen 2.5 se muestra un ejemplo de la identificación de caracteres.

	h_1	h_2	h_3	h_4	h_5	h_6	h_7
A	2.837e−1	1.961e−3	1.484e−2	2.265e−4	−4.152e−7	1.003e−5	−7.941e−9
I	4.578e−1	1.820e−1	0.000	0.000	0.000	0.000	0.000
O	3.791e−1	2.623e−4	4.501e−7	5.858e−7	1.529e−13	7.775e−9	−2.591e−13
M	2.465e−1	4.775e−4	7.263e−5	2.617e−6	−3.607e−11	−5.718e−8	−7.218e−24
F	3.186e−1	2.914e−2	9.397e−3	8.221e−4	3.872e−8	2.019e−5	2.285e−6

Figura 2.5: Ejemplos de los valores de los momentos invariantes de *Hu* para diferentes caracteres.

2.5.1 Descriptores de profundidad para reconocimiento de objetos

Como objetivo de este trabajo se han estudiado varios descriptores de imágenes en profundidad, para obtener una mejor visión de las nuevas posibilidades que se pueden emplear para extraer características de los objetos evaluados.

En esta sección se va a estudiar unos descriptores de imágenes en profundidad realizados por [7] en los cuales se han obtenido buenos resultados para la detección de diversos objetos. En [7] se han obtenido descriptores tanto para imágenes en RGB como de profundidad, en nuestro caso se analizan los descriptores de información en profundidad.

2.5.1.1 Depth kernel descriptor

Uno de los enfoques más comunes para el reconocimiento de objetos consiste en obtener las características de los píxeles que forman la imagen considerando pequeñas subregiones formadas por un conjunto de puntos, a las cuales llamaremos ventanas. Un tamaño común en la actualidad suele ser ventanas de 5x5 píxeles. El tamaño de la ventana empleada tiene una gran importancia ya que puede suponer una ocultación de ciertos píxeles discriminantes ante la presencia de otros.

Los descriptores tipo *Kernel* evitan la discretización de los atributos de los píxeles adoptando una visión similar a un parche. En este caso la similitud entre dos parches se basa en una función *Kernel*, conocida como *match kernel*, que proporciona un promedio de las similitudes entre los píxeles de los dos parches. Los *match kernel* son extremadamente flexibles, ya que la función *kernel* empleada para evaluar los atributos de los píxeles puede ser desde una función gaussiana, hasta las mas conocidas SIFT, [8], o HOG, [9].

Mientras que los *match kernel* proporcionan una medida de similitud natural para las ventanas que se emplean, su procesamiento puede resultar muy costoso cuanto mayor sea la ventana.

Aun tras la selección de una ventana de dimensiones reducidas, se suelen obtener vectores de características sumamente grandes, por lo que es necesario reducir su dimensión empleando el análisis de sus componentes principales, como se comentará en la sección 2.6.2.

2.5.1.2 Tamaño de la ventana de pixeles

Los descriptores de imágenes de profundidad son sensibles al tamaño de la región de interés que se desee evaluar, por ello se ha de evaluar el tamaño de estas regiones. En esta sección se estudia el tamaño del *kernel* empleado, para ello se debe estudiar la distancia entre cada punto de la nube y el punto de referencia de la nube de puntos, P denota la nube de puntos evaluados, $\bar{p}$ representa el punto de referencia de tal nube, p representa un punto perteneciente a la nube P y d_p representa la distancia entre el punto p y el punto de referencia de la nube $\bar{p}$.

Para evaluar la similaridad entre los atributos de distancia de dos nubes de puntos, P y Q se emplea el *Kernel* de tamaño K_{size}

$$d_p = \| p - \bar{p} \|_2 \; \forall p \in P \tag{2.19}$$

$$K_{size}(P, Q) = \sum_{p \in P} \sum_{q \in Q} k_{size}(d_p, d_q) \tag{2.20}$$

Donde $k_{size}(d_p, d_q)$ es una *kernel* formado por una función *gaussiana*, la cual caracteriza la similitud entre las dos nubes de puntos, basándose en las distancias entre los elementos que los forman.

$$k_{size}(d_p, d_q) = exp(-\gamma_s \| d_p - d_q \|_2)(\gamma_s > 0) \tag{2.21}$$

Al introducir el *kernel gaussiano*, la dimension del vector de características de la nube de puntos P resulta tener dimensión infinita, por lo que es necesario proyectar tales infinitas dimensiones sobre un conjunto finito de vectores, los cuales representan las características principales obtenidos por la técnica PCA (*Principal Components Analysis*), [10], dando lugar a un vector de características de tamaño finito, obtenido por el descriptor *kernel* de dimensiones finitas, $F^e_{size}(P)$.

$$F^e_{size}(P) = \sum_{t=1}^{b_s} \alpha^e_t \sum_{p \in P} k_{size}(d_p, u_t) \tag{2.22}$$

Donde u_t son los vectores básicos que representan los atributos de distancia desde la región de soporte, b_s es el número de vectores básicos, y ${\alpha^e}_{e=1}^{E}$ son los E autovectores calculados en el análisis de sus componentes principales.

2.5.1.3 Forma de los objetos evaluados

Los objetos evaluados tienen una forma 3D considerablemente constante por lo que para su caracterización se emplea como base principal su forma. Para ello se centra el estudio en 2 aspectos: las características del análisis de sus componentes principales y el *spin kernel descriptor*.

Con el análisis del *kernel* K_p sobre la nube de puntos P y evaluando los L autovalores obtenidos en el análisis de sus componentes principales, con lo que se obtiene las características locales del *kernel PCA*, $[\lambda^1_p, \ldots, \lambda^l_p, \cdots, \lambda^L_p]$ con:

$$K_p v^l = \lambda^l_p v^l \tag{2.23}$$

Donde v^l son los autovectores, L es la dimensionalidad local tras el análisis de sus componentes principales, y $K_p[s,t] = exp(-\gamma_k)$

La obtención descriptores basados en el *spin* (orientación) es muy empleada en la actualidad para el reconocimiento de patrones. Las nubes de puntos 3D se representan por el par $(\bar{p}, \bar{n})$, donde $\bar{p}$ representa sus coordenadas 3D y $\bar{n}$ representa la superficie normal. Y en el caso de un punto $p \in P$, viene representado por el par (p, n). A partir de los cuales se pueden obtener $[\eta_p, \zeta_p, \beta_p]$:

$$\eta_p = n \times (p - \bar{p}) \tag{2.24}$$

$$\zeta_p = \sqrt{\| p - \bar{p} \|^2 - \eta_p^2} \tag{2.25}$$

$$\beta_p = arccos(n \times \bar{n}) \tag{2.26}$$

Donde la coordenada de elevación η_p viene definida por la distancia perpendicular desde el punto p hasta la tangente del plano definido por el par de posición y normal $(\bar{p}, \bar{n})$, la coordenada radial ζ_p es la distancia perpendicular desde el punto p hasta la linea a través de la normal $\bar{n}$, y β_p es el angulo entre las normales n y $\bar{n}$.

Agregando los atributos del punto $[\eta_p, \zeta_p, \beta_p]$ en las características locales de forma, dando lugar al siguiente *spin kernel*.

$$K_{spin}(P) = \sum_{p \in P} \sum_{q \in Q} k_a(\bar{\beta_p}, \bar{\beta_q}) k_{spin}([\eta_p, \zeta_p], [\eta_q, \zeta_q]) \tag{2.27}$$

Donde $\bar{\beta_p} = [sin(\beta_p), cos(\beta_p)]$, P es el conjunto de puntos cercanos al punto de referencia $\bar{p}$. Los *kernels* gaussianos k_a y k_{spin} miden la similaridad entre los atributos β, η y ζ respectivamente.

Obteniendo al igual que en el caso del *size kernel*, K_{size}, un *kernel* de tamaño finito gracias al empleo de la técnica PCA para la obtención y posterior proyección sobre sus componentes principales.

2.5.1.4 Obtención de información de los bordes

Para poder evaluar los bordes de las regiones de interés, se necesita convertir la nube de puntos 3D en una imagen bidimensional en escala de grises, donde el valor de la intensidad de cada uno de os píxeles equivaldrá al valor de su profundidad.

Por lo que tratándose de mapas de píxeles en 2D se pueden aplicar gradientes y patrones locales al mapa de puntos para la extracción de información de sus bordes. De tal forma que se obtiene el *gradient match kernel*, K_{grad}, a partir de los atributos de gradiente de los píxeles.

$$K_{grad}(P, Q) = \sum_{p \in P} \sum_{q \in Q} \tilde{m_p} \tilde{m_q} k_o(\tilde{\theta_p}, \tilde{\theta_q}) k_s(p, q) \tag{2.28}$$

Donde P y Q son parches de diferentes imágenes, y $p \in P$ es la posición 2D de un píxel correspondiente a un parche de profundidad normalizado entre [0,1]. θ_p y m_p son la orientación y la magnitud del gradiente de profundidad del píxel p. Los *normalized linear kernel* $\tilde{m_p}\tilde{m_q}$ representan el peso de la contribución de cada gradiente.

$$\tilde{m_p} = \frac{m_p}{\sqrt{\sum_{p \in P} m_p^2 + \varepsilon_g}} \tag{2.29}$$

Donde ε_g es una constante pequeña y positiva que el denominador es mayor que 0. El *position gaussian kernel*, k_s, mide como de cerca se encuentran dos píxeles espacialmente, el *orietation kernel*, k_o, evalúa la similaridad de la orientación de los gradientes, donde $\tilde{\theta_p} = [sin(\theta_p), cos(\theta_p)]$.

$$k_s(p,q) = exp(-\gamma_s \parallel p - q \parallel^2) \tag{2.30}$$

$$k_o(\tilde{\theta_p}, \tilde{\theta_q}) = exp(-\gamma_o \parallel \tilde{\theta_p} - \tilde{\theta_q} \parallel^2) \tag{2.31}$$

El *local binary kernel descriptor*, K_{lbp} viene descrito a continuación, donde s_p es la desviación estándar alrededor de un píxel p con nivel de vecindad $3x3$, ε_{lbp} es una constante pequeña y positiva que asegura que el denominador sea mayor que cero.b_p es un vector columna binario que binariza las diferencias de los valores de los píxeles en una ventana local alrededor del punto p. k_b es un *gaussian kernel* que mide las similaridades locales entre los parches binarios locales.

$$\tilde{s_p} = \frac{s_p}{\sqrt{\sum_{p \in P} s_p^2 + \varepsilon_{lbp}}} \tag{2.32}$$

$$k_b(b_p, b_q) = exp(-\gamma_b \parallel b_p - b_q \parallel^2) \tag{2.33}$$

$$K_{lbp}(P,Q) = \sum_{p \in P} \sum_{q \in Q} \tilde{s_p} \tilde{s_q} k_b(b_p, b_q) k_s(p,q) \tag{2.34}$$

Obteniendo al igual que en el caso del *kernel* del tamaño, K_{size}, un *kernel* de tamaño finito gracias al empleo de la técnica PCA para la obtención y posterior proyección sobre sus componentes principales, dado que sino se obtendría un *kernel* de tamaño infinito.

2.5.1.5 Obtención de *match kernel* de forma piramidal

También se emplean EMK (*Efficient Match Kernels*) de forma piramidal para agregar *kernels* locales en los diferentes niveles de características. Esta técnica robustece el sistema combinando las ventajas de los anteriores *kernels*, al obtener los descriptores locales en un espacio de puntos de características de baja dimensión y empleando la construcción de los niveles de características del objeto a partir de la media de los vectores resultantes. Los EMK que se emplean se muestran en la siguiente ecuación:

$$K_{emk}(P,Q) = \sum_{p \in P} \sum_{q \in Q} k_f(p,q) = \bar{\phi_{emk}}(P)^T \bar{\phi_{emk}}(Q) \tag{2.35}$$

Donde P y Q son conjuntos de descriptores de *kernel* locales que representan imágenes en profundidad de objetos, $\bar{\phi_{emk}}$ es el vector de características de la imagen en profundidad de un objeto. Tras ello se deriva el *finite dimensional kernel* k_f a partir del *kernel* gaussiano en 2 pasos:

- Se estudian un conjunto de vectores base mediante la técnica k-means sobre un gran grupo de descriptores locales de *kernel*.
- Se diseña un *kernel* de dimensiones finitas a partir de las proyecciones de las infinitas dimensiones de las características introducidas por el *kernel* gaussiano, empleando un *kernel* restringido con valores singulares.

La información espacial es incorporada en los EMK de forma similar a un *matching* espacial-piramidal. Concatenando las características EMK de diferentes regiones se obtiene las características del EMK piramidal.

$$K_{emk}(P,Q) = \sum_{p \in P} \sum_{q \in Q} k_f(p,q) = \bar{\phi_{emk}}(P)^T \bar{\phi_{emk}}(Q) \tag{2.36}$$

Donde R es el numero de niveles piramidales, T es el numero de celdas espaciales en cada r nivel piramidal. $\bar{\phi_{emk}}(P^(R, T_R))$ con las características del *efficient match kernel* comprendidas dentro de la región espacial $P^(R, T_R)$.

Los mapas de características piramidales son aplicados sobre objetos para la obtención de sus características en niveles del objeto. La dimensionalidad de las características del *pyramid efficient match kernel* depende del producto de número de vectores básicos y en número de subregiones formadas por la descomposición piramidal, la cual ha de ser igual en diferentes objetos. Gracias a los *pyramid efficient match kernel* es posible obtener una transformación rápida y no lineal sobre los descriptores de *kernel* locales.

2.5.2 Descriptores de profundidad para la detección de personas en imágenes cenitales

Los descriptores que se tratan en este apartado están destinados a la detección de personas dentro de un entorno concreto, dado que se desea tener el menor número de oclusiones, se posiciona la cámara en posición cenital. Para detectar a las personas que se encuentran dentro del entorno se tiene en cuenta la morfología de las mismas, es decir, la fisiología de su cabeza cuello y hombros, ya que las demás características de las personas pueden quedar ocultas debido a la posición cenital. Los vectores de características obtenidos, se basan en la densidad de puntos que se encuentran en diferentes niveles de altura que se definirán más adelante, por lo que la magnitud de cada característica variará con la fisonomía de la persona.

El vector de características obtenido cuenta con 6 componentes, cinco de ellas dependen de la situación de los puntos de la imagen tridimensional dentro de los niveles marcados, y la última componente indica el grado de excentricidad de la cabeza de la posible persona detectada, es decir la relación entre los diámetros mayor y menor de la superficie superior.

En el entorno evaluado se pueden encontrar personas, animales u objetos, todos ellos forman parte del grupo de candidatos a ser evaluados por el sistema. Para ello se analizan las medidas encontradas en la escena, para formar un conjunto robusto de máximos que tenga en cuenta todos los candidatos. Alrededor de los máximo se seleccionan las regiones de interés (ROI) que se han de evaluar para determinar posteriormente si se trata de una persona o no, como se trata en [5]. La ROI a evaluar consiste en una nube de puntos tridimensional, de la cual se pueden diferenciar puntos pertenecientes a la cabeza, cuello u hombros. Los puntos que se evalúan cumplen la condición de pertenecer a la región comprendida entre el máximo detectado, al que se llamará altura máxima detectada, h_{max} y una distancia de interés, definida como ΔH, la cual por procesos experimentales se determina: $\Delta H = 40cm$. Con $40cm$ se asegura que, en caso de tratarse de una persona, todos los puntos correspondientes a cabeza, cuello y hombros se encuentran en la ROI. La región evaluada se fragmenta en franjas de altura, Δh, comprendiendo cada una de ellas $2cm$, de tal forma se obtendrán 20 franjas de altura idéntica para evaluar, como se observa en la figura 2.6.

Se continúa evaluando la densidad de puntos encontrados en cada una de las franjas, de tal forma que se obtiene un vector de 20 componentes, donde la magnitud de cada uno de ellos consiste en la cantidad de puntos que se encuentran en la franja correspondiente. El vector obtenido tiene una gran variabilidad debido a que la posición cenital de la cámara pude dar lugar a la oclusión de ciertas zonas del cuerpo en

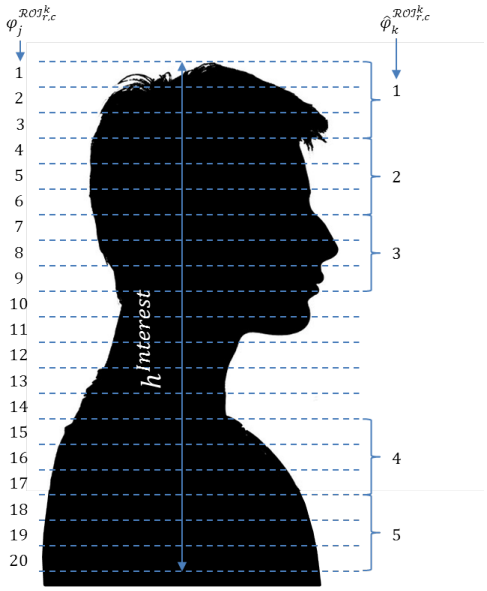

Figura 2.6: Esquema de las diferentes subregiones de idéntica altura que fragmentan la región evaluada.

función de la posición de la persona en la escena, así como la fisonomía de la persona, por ejemplo pelo largo o una coleta que limite los puntos encontrados en los hombros. También hay que considerar el ruido encontrado en la imagen así como los efectos del multicamino de las cámaras ToF.

Para obtener un sistema más robusto y fiable se deben minimizar los errores debidos a los efectos comentados, por lo que se reduce el tamaño del vector de características de 20 a 6 componentes, dentro de las cuales las 5 primeras corresponderán a la densidad de puntos, y el último a la excentricidad de la cabeza de la posible persona detectada. A continuación se muestra el procedimiento que se ha empleado para la obtención de las componentes mostradas en la siguiente ecuación.

$$\varphi^{(}\hat{ROI}_{r,c}^k) = \{\varphi_1^{(}\hat{ROI}_{r,c}^k), \ldots, \varphi_{N_\Phi}^{(}\hat{ROI}_{r,c}^k)\} \tag{2.37}$$

Donde r y c representan las coordenadas pixélicas de la ROI evaluada dentro del plano imagen, k es un indice que recorre los máximos de la imagen y N_Φ es el número de componentes del vector de características, 6 en este caso.

2.5.2.1 Componentes primera, segunda y tercera

Estas componentes representan la información de la cabeza, de tal forma que cada una de las tres componentes esta formada por el los puntos acumulados en 3 franjas consecutivas. La primera franja evaluada dependerá de cual de las 3 primeras subregiones contenga mas puntos acumulados, gracias a lo cual, se evita que ante una medida errónea del máximo de la posible persona evaluada de lugar a una detección errónea de la persona.

Para la determinación de la zona con mayor densidad de puntos se emplea la siguiente ecuación:

$$\mu_H = argmax_{1\leq s\leq 3}\{\varphi_s^{(}ROI_{r,c}^k)\} \tag{2.38}$$

A partir de este valor se evalúan las tres primeras componentes como se muestra a continuación:

$$if \mu_H = 1 \quad OR \quad \mu_H = 2 \Rightarrow \begin{cases} \hat{\varphi}_1^{ROI_{r,c}^k} & = & \sum_{s=1}^{3} \varphi_s^{ROI_{r,c}^k} \\ \hat{\varphi}_2^{ROI_{r,c}^k} & = & \sum_{s=4}^{6} \varphi_s^{ROI_{r,c}^k} \\ \hat{\varphi}_3^{ROI_{r,c}^k} & = & \sum_{s=7}^{9} \varphi_s^{ROI_{r,c}^k} \end{cases} \tag{2.39}$$

$$if \mu_H = 3 \Rightarrow \begin{cases} \hat{\varphi}_1^{ROI_{r,c}^k} & = & \sum_{s=2}^{4} \varphi_s^{ROI_{r,c}^k} \\ \hat{\varphi}_2^{ROI_{r,c}^k} & = & \sum_{s=5}^{7} \varphi_s^{ROI_{r,c}^k} \\ \hat{\varphi}_3^{ROI_{r,c}^k} & = & \sum_{s=8}^{10} \varphi_s^{ROI_{r,c}^k} \end{cases} \tag{2.40}$$

2.5.2.2 Componentes cuarta y quinta

Estas componentes representan la información de los hombros, donde cada componente es formada a partir de 3 subregiones de las 20 iniciales. La selección de las subregiones que forman parte de estas dos componentes, se determina en función de cuál de las últimas 10 subregiones contiene un mayor número de puntos 3D, es decir cuál de las últimas 10 características de vector inicial de 20 tiene una mayor densidad de puntos, μ_S. Con ello se evita la selección de regiones pertenecientes al cuello que por lo general no contiene información relevante debido a la situación cenital en la que se encuentra la cámara.

$$\mu_S = argmax_{10 \leq s \leq 20}\{\varphi_s^(ROI_{r,c}^k)\} \tag{2.41}$$

$$\hat{\varphi}_4^{ROI_{r,c}^k} = \sum_{s=\mu_S-1}^{\mu_S+1} \varphi_s^{ROI_{r,c}^k} \tag{2.42}$$

$$\hat{\varphi}_5^{ROI_{r,c}^k} = \sum_{s=\mu_S-1}^{\mu_S+1} \varphi_s^{ROI_{r,c}^k} \tag{2.43}$$

Como se describe en los apartados anteriores las franjas a partir de las cuales se forma el vector de características no son fijas, sino que dependen de la fisonomía de la persona que se este evaluando, de tal forma que será más personalizado conforme a la persona. En la imagen 2.7 se puede observar un ejemplo de las las 20 franjas iniciales a partir de las cuales se seleccionan las mas importantes para su estudio, así como las que forman el vector de características de la persona en cuestión.

2.5.2.3 Normalización de las cinco primeras componentes

En los apartados anteriores se han obtenido las 5 primeras componentes del vector de características, pero no se ha tenido en cuenta un factor sumamente importante, la altura de la persona, ya que en función de la tal magnitud, la región que ocupe dentro del entorno tendrá una mayor densidad de píxeles. Una persona con altura menor tendrá una región menos poblada que una persona de mayor altura, por tanto es necesario normalizar las 5 primeras componentes del vector de características conforme a la altura de la persona.

Para llevar a cabo la normalización se parte de dos datos determinantes, el numero de píxeles que se encuentran en la primera franja de la cabeza, es decir la primera componente del vector de características, $\hat{\varphi}_1^{ROI_{r,c}^k}$, y la altura $\hat{h}_{r,c}^{maxSR}$, la cual determina la altura máxima de la persona que se esta evaluando.

Una vez obtenidas estas magnitudes se procede a estimar la relación existente entre ambas, asumiendo en primer lugar una relación cuadrática, como muestra la ecuación 2.44

$$\hat{\varphi}_1^{ROI_{r,c}^k} \approx \rho_1^{ROI_{r,c}^k} = a_0(\hat{h}_{r,c}^{maxSR})^2 + a_1\hat{h}_{r,c}^{maxSR} + a_2 \tag{2.44}$$

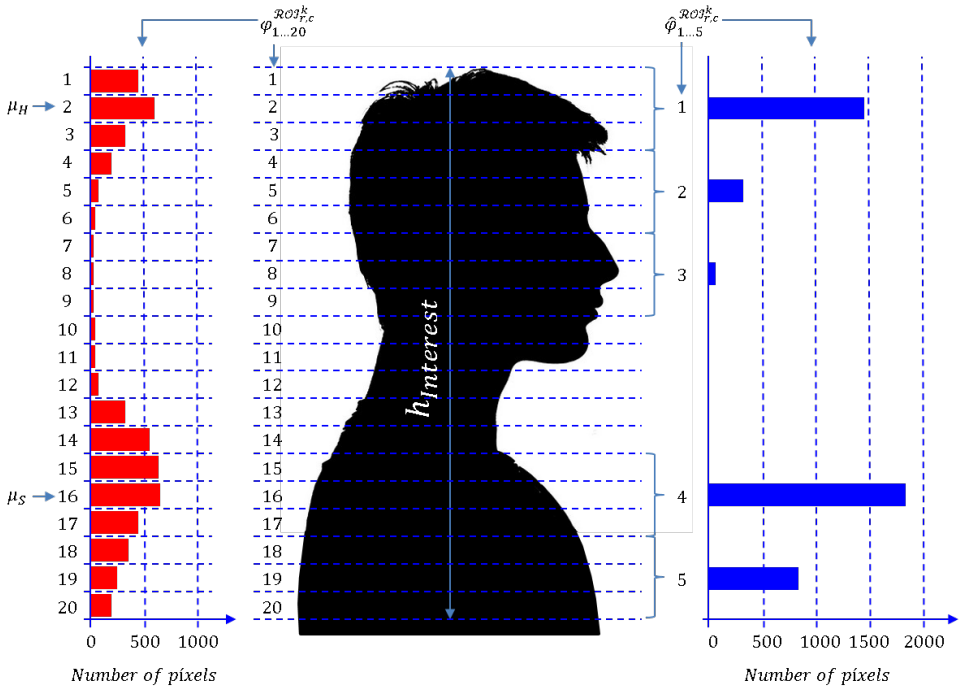

Figura 2.7: Esquema de las diferentes subregiones de idéntica altura que fragmentan la región evaluada.

Donde a_0 , a_1 y a_2 son los coeficientes que se han de evaluar. Para su obtención se emplea el algoritmo Levenberg-Marquardt para que se ajusten a los datos de entrada. De forma experimental se observó que para alturas de personas comprendidas entre 140cm y 213cm. Tras ello se obtienen los coeficientes indicados en la ecuación 2.45. Para estos valores, el error cuadrático medio es de aproximadamente 45 píxeles, ver imagen 2.8.

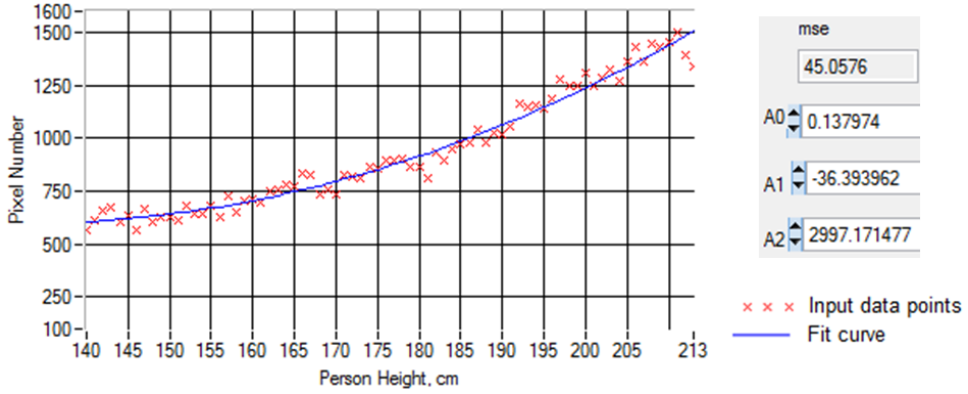

Figura 2.8: Gráfica de $\rho_1^{ROI^k_{r,c}}$, donde se muestra el conjunto de datos de entrenamiento y la curva ajustada, así como los valores del conjunto de coeficientes a_0 , a_1 y a_2 y el error cuadrático medio obtenido.

$$\begin{cases} a_0 = 0{,}137974 \\ a_1 = -36{,}393962 \\ a_1 = 2997{,}171477 \end{cases} \tag{2.45}$$

Finalmente se obtienen las 5 primeras componentes del vector de características normalizadas al dividir las componentes calculadas, $\hat{\varphi}^{ROI^k_{r,c}}$, entre la estimación $\rho_1^{ROI^k_{r,c}}$.

$$\hat{\varphi}_i^{ROI_{r,c}^k} = \frac{\hat{\varphi}_i^{ROI_{r,c}^k}}{\rho_1^{ROI_{r,c}^k}}, \quad \text{con } 1 \leq \text{i} \leq 5 \tag{2.46}$$

2.5.2.4 Componente sexta

Dado que las 5 componentes obtenidas anteriormente solo introducen información sobre la densidad de puntos comprendidos en las diferentes franjas del individuo evaluado, es necesario aportar información extra para poder obtener un vector de características mas fiable y robusto. Por ello se incorpora la sexta componente, que da información a cerca de la forma en la que se distribuyen los píxeles que forman parte de la zona superior de la cabeza, es decir, los píxeles comprendidos entre $h_{r,c}^{maxSR}$ y $h_{r,c}^{maxSR}$ - 6 cm.

Por tanto la sexta componente, $\hat{\varphi}_6^{ROI_{r,c}^k}$, dará información a cerca de la fisiología de la cabeza, sobre su circularidad.

$$\hat{\varphi}_6^{ROI_{r,c}^k} = \text{rba}(h_{r,c}^{maxSR} \geq Z_{i,j} > h_{r,c}^{maxSR} - 3\Delta h) \tag{2.47}$$

Donde *rba* indica el ajuste al menor sesgo.

2.6 Clasificadores

En la gran mayoría de los sistemas de visión artificial es necesario diferenciar los elementos que se encuentran en el entorno, para ello se recurre a los clasificadores, que tienen como principal objetivo asociar elementos a clases o grupos ya identificados. De forma habitual los clasificadores necesitan que los elementos estén representados por vectores de características discriminantes entre sí. Una de las dificultades mas frecuentes que se pueden encontrar en estos sistemas es la variabilidad de las componentes de los vectores de características, el cual se puede producir por el ruido del entorno o el sistema de adquisición así como de la dependencia o independencia entre las componentes del vector de características.

En la actualidad existen multitud de clasificadores, desde los mas simples que discriminan por la distancia, hasta clasificadores mas sofisticados como PCA (Principal Components Analysis), SVM (Support Vector Machines), redes neuronales, ramdon forest, etc.

Se pueden diferenciar diversos tipos de clasificadores, dependiendo de diferentes parámetros:

- Clasificadores supervisados: en los cuales se conocen la totalidad de las clases de antemano.
- Clasificadores no supervisados: en los cuales no se conocen ni el número de clases ni las clases en sí.
- Clasificadores a priori: son aquellos clasificadores que se construyen en un único paso a partir de unos datos de muestra previamente proporcionados.
- Clasificadores a posteriori: son aquellos clasificadores que se construyen en un proceso continuo e iterativo, gracias al cual el clasificador reconoce los datos de muestra previamente proporcionados de forma iterativa.
- Clasificadores deterministas: son aquellos clasificadores en los cuales ante la entrada de los mismos vectores de características y con la misma configuración el clasificador da la misma salida.
- Clasificadores no deterministas: son aquellos clasificadores en los cuales la salida no permanece contante ante la misma entrada de vectores de característica, ya que depende de otros factores, por ejemplo probabilísticos.

- Clasificadores monoclase: clasificadores en los cuales solo se discrimina la pertenencia o no se un ítem a una clase concreta y única.
- Clasificadores multiclase: son aquellos clasificadores en los cuales un vector de características puede asociarse a 2 o mas clases.

A continuación se describen diversos clasificadores que se han empleado en este Trabajo Fin de Grado.

2.6.1 Maquina de soporte vectorial

Las máquinas de soporte vectorial (SVM) son clasificadores de vectores de características basados en diferentes técnicas, una única clase, separación por hiperplanos, separación con posibles imperfecciones, etc. Estos clasificadores fueron desarrollados en los años 90 por Vapnik y sus colaboradores, [11] y [12], en primer lugar las SVM fueron desarrolladas para resolver problemas de clasificación binaria, aunque actualmente se emplean para la resolución de diversos problemas como multiclase, regresión, etc.

Las SVM se clasifican como supervisados debido a que a partir de un número determinado de muestras, caracterizadas por vectores de características con un número de características determinado, un número de clases concretas, así como de la pertenencia de las muestras a las clases, se crea un modelo a partir del cual se podrá predecir en posteriores iteraciones la clase de pertenencia de un vector de características desconocido. Las fronteras de decisión entre las diferentes clases suelen ser flexibles en estos clasificadores.

Las máquinas de soporte vectorial realizan clasificaciones lineales, ya que proporcionan separadores lineales o hiperplanos para diferenciar las posibles clases. En el caso de que en el espacio original no sean separables linealmente se realiza una transformación hasta llegar a un espacio en el que se dé una separación lineal. Para la realización de las transformaciones se emplean las funciones denominadas *Kernel*.

La idea principal de las SVM es minimizar los errores cometidos en el modelo generado a partir de la muestra de entrenamiento. Por ello se selecciona un hiperplano que equidiste de los ejemplos mas cercanos de cada clase, gracias a ello se obtiene un margen máximo entre el hiperplano y cada clase. Para seleccionar el hiperplano únicamente se consideran la muestras de entrenamiento que quedan en la frontera de tales márgenes, estas muestras reciben el nombre de *vectores soporte*.

2.6.1.1 Clasificadores SVM con separación lineal

Como explica [13] se puede lograr la clasificación de un conjunto de puntos con separación lineal. Dado un conjunto separable de ejemplos $S = (x_1, y_1), \ldots, \ldots (x_n, y_n)$, donde $x_i \in \mathbb{R}^d$, se puede definir un hiperplano de separación como una función lineal que es capaz de separar dicho conjunto sin error:

$$D(x) = (w_1 x_1 + \ldots + w_d x_d) + b =< w, x > +b \tag{2.48}$$

Donde w y b son coeficientes reales. El hiperplano de separación cumplirá las siguientes restricciones para todo x_i:

$$\begin{cases} < w, x_i > +b \geq 0 \text{si} y_i = +1 & \text{con i=1,...,n} \\ < w, x_i > +b \leq 0 \text{si} y_i = -1 & \text{con i=1,...,n} \end{cases} \tag{2.49}$$

Es decir,

$$y_i D(x_i) \geq 0 \quad \text{con i=1,. . .,n} \tag{2.50}$$

El hiperplano que permite separar los conjuntos no es único, como se puede ver en la imagen 2.9 existe infinitos planos que permiten separarlos. Para seleccionar el plano optimo se introduce el concepto de *margen* de un hiperplano de separación, al que se denotara por τ, el cual sera la mínima distancia entre el hiperplano y la muestra más cercana de cualquiera de las clases. De esta forma un hiperplano sera óptimo si su margen es el máximo posible. Debido a esta característica se afirma que un hiperplano óptimo equidista de la muestra mas cercana de cada clase.

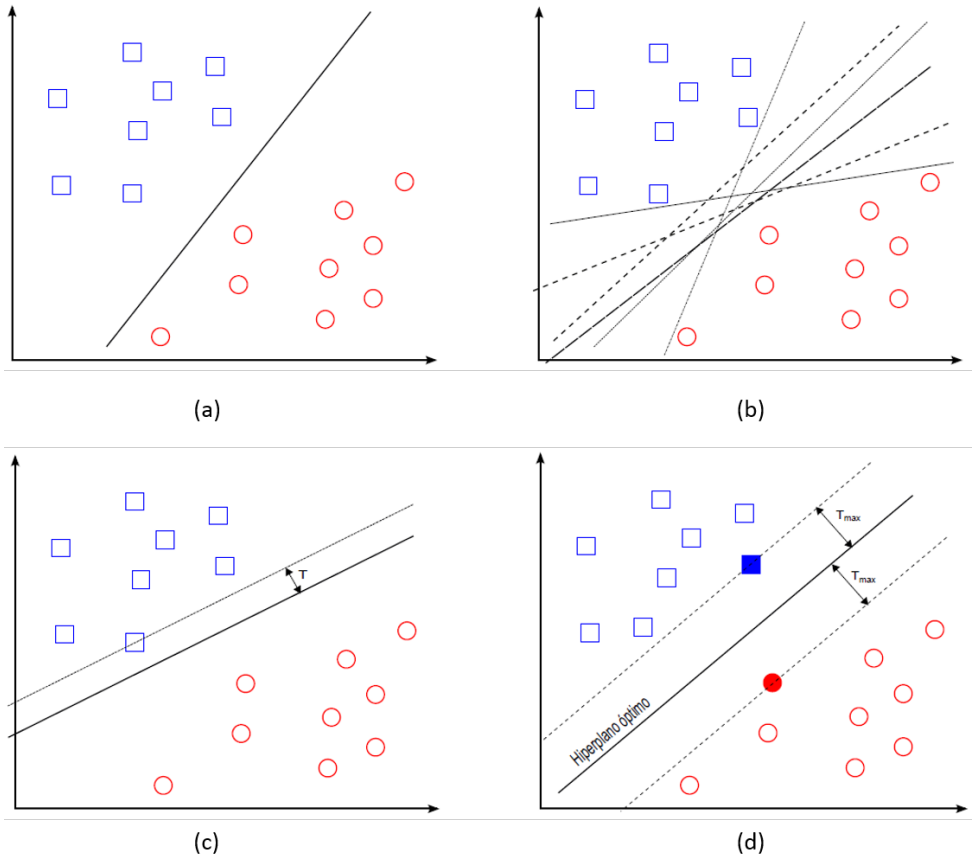

Figura 2.9: (a)Ejemplo de hiperplano de separación. (b) Diferentes ejemplos de hiperplanos de separación que muestran las infinitas posibilidades. (c) Ejemplo de hiperplano no óptimo con margen no máximo. (d) Ejemplo de hiperplano óptimo con margen de separación máximo

Por geometría se conoce que la distancia entre un hiperplano de separación *D(x')* y un ejemplo *x'* viene dada por:

$$\frac{|D(x')|}{\|w\|} \tag{2.51}$$

Por tanto todos los ejemplos de entrenamiento empleados para la generación del hiperplano han de cumplir la condición de la ecuación 2.52

$$\frac{y_i D(xi)}{\|w\|} \geq \tau \quad \text{con i=1,. . .,n} \tag{2.52}$$

De tal forma que encontrar el hiperplano óptimo es equivalente a encontrar el valor de w que maximiza

el margen, pero existen infinitas soluciones que difieren solo en la escala de w, por ello, para limitar el número de soluciones a una sola se fija la escala del producto de τ y la norma de w a la unidad.

$$y_i D(xi) \geq \tau \|w\| \quad \text{con i=1,. . .,n} \tag{2.53}$$

$$\tau \|w\| = 1 \tag{2.54}$$

$$\tau = \frac{1}{\|w\|} \tag{2.55}$$

Por lo que se concluye que disminuir la norma de w equivale a aumentar el margen τ. Un hiperplano de separación óptimo se caracterizara por poseer un margen máximo, y un valor mínimo de la norma de w, es decir, cuanto mayor sea el margen mayor sera la distancia de separación que exista entre las dos clases.

$$y_i(< w, x_i > +b) \geq 1 \quad \text{con i=1,. . .,n} \tag{2.56}$$

Los elementos situados a ambos lados del hiperplano óptimo que definen el margen, reciben el nombre de *vectores soporte*, y ellos solos definen el hiperplano de separación optimo.

La búsqueda del hiperplano óptimo para el caso separable linealmente consiste en la obtención de el valor de w y b, que cumple las condiciones de las ecuaciones 2.57 y 2.58.

$$min \quad f(w) = \frac{1}{2}\|w\|^2 = \frac{1}{2} < w, w > \tag{2.57}$$

$$s.a. \quad y_i(< w, x_i > +b) - 1 \geq 0 \quad \text{con i=1,. . .,n} \tag{2.58}$$

Para la resolución de este problema de optimización con restricciones, se emplea la *teoría de la optimización*. Se comienza por construir un problema de optimización sin restricciones empleando la función Lagrangiana:

$$L(w, b, \alpha) = \frac{1}{2} < w, w > - \sum_{i=1}^{n} \alpha_i [y_i(< w, x_i > +b) - 1] \tag{2.59}$$

Donde $\alpha_i \geq 0$ son los multiplicadores de *Lagrange*. Tras este primer paso, se procede con la aplicación de las condiciones de *Karush-Kuhn-Tucker*, también conocidas como las condiciones KKT:

$$\frac{\partial L(w^*, b^*, \alpha)}{\partial w} \equiv w^* - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \quad \text{con i=1,. . .,n} \tag{2.60}$$

$$\frac{\partial L(w^*, b^*, \alpha)}{\partial b} \equiv \sum_{i=1}^{n} \alpha_i y_i = 0 \quad \text{con i=1,. . .,n} \tag{2.61}$$

$$\alpha_i [1 - y_i(< w^*, x_i > +b^*)] = 0 \quad \text{con i=1,. . .,n} \tag{2.62}$$

Gracias a las condiciones KKT, se pueden expresar los parámetros w y b en referenciadas a los multiplicadores de *Lagrange*, α_i, y además establecen restricciones para tales términos.

$$w^* = \sum_{i=1}^{n} \alpha_i y_i x_i \quad \text{con i=1,. . .,n} \tag{2.63}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \quad \text{con i=1,...,n} \tag{2.64}$$

Con ello se construirá el problema dual, en el cual se ha pasado de un problema inicial de minimización primal, a un problema de maximización de la ecuación 2.65 pero respetando las restricciones asociadas a los multiplicadores de Lagrange , ecuación 2.66.

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j < x_i, x_j > \tag{2.65}$$

$$\begin{cases} max \quad L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j < x_i, x_j > \\ s.a. \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \\ s.a. \quad \alpha_i \geq 0 \quad \text{con i=1,...,n} \end{cases} \tag{2.66}$$

La gran ventaja del problema dual consiste en la reducción del coste computacional, aun con problemas con un numero alto de dimensiones. Tras su solución, podemos obtener la solución al problema primal sustituyendo α^* en la ecuación 2.67

$$D(x) = \sum_{i=1}^{n} \alpha_i^*, y_i < x, x_i > + b^* \tag{2.67}$$

Los *vectores de soporte* son aquellos ejemplos que cumplen las restricciones mencionadas, por ello solo ellos cumplirán la condición $\alpha_i > 0$. Dado que el hiperplano óptimo se construirá con todos los vectores de soporte, estos serán empleados para la determinación de b^*, pudiendo emplear la tupla de un vector de soporte, (x_{vs}, y_{vs}), junto con su valor de clase, ecuación 2.68, o empleando un promedio de los N_{vs} vectores de soporte, ecuación 2.69.

$$b^* = y_{vs} - < w^*, x_{vs} > \tag{2.68}$$

$$b^* = \frac{1}{N_{vs}} \sum_{i=1}^{N_{vs}} (y_{vs} - < w^*, x_{vs} >) \tag{2.69}$$

2.6.1.2 Clasificadores SVM con separación cuasi-lineal

La estrategia que se emplea en este tipo de problemas reales consiste en relajar el grado de separabilidad del conjunto de datos, permitiendo errores de clasificación en algunos ejemplos del conjunto de datos de entrenamiento. Por ello se pueden dar dos casos, y en ambos se dice que el ejemplo es no-separable:

- Clasificación correcta: el ejemplo que se clasifica cae dentro del margen correspondiente a la clase correcta.
- Clasificación incorrecta: el ejemplo que se clasifica no cae dentro del margen de la clase correcta, es decir al otro lado del hiperplano que define la separación.

Para abordar este problema se van a introducir un conjunto de variables reales positivas, que permitirán cuantificar el número de ejemplos no-separables que se permitirán, estas variables se denominaran *variables de holgura*, $\xi_i, i = 1, ..., n$, como se muestra en la ecuación 2.70.

$$y_i(< w, x_i > + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \text{ i=1,...,n} \tag{2.70}$$

Para el ejemplo anterior, (x_i, y_i), su variable de holgura, ξ_i representa la desviación del caso separable, comenzando a medir desde el límite del margen que corresponde a la clase y_i, por lo que si su variable de holgura es igual a cero, el ejemplo es separable, si es mayor que cero pero menor que uno, el ejemplo es no separable, y finalmente si es mayor que uno estamos ante un ejemplo no separable y mal clasificado. Por lo que si obtenemos la suma de todas las variables de holgura podemos obtener una idea de el número de ejemplos no separables, siendo siendo estos dos valores directamente proporcionales, es decir, cuanto mayor sea el sumatorio de las variables de holgura, mayor sera el número de ejemplos no separables. En la imagen 2.10 se puede observar gráficamente la holgura de los diferentes ejemplos.

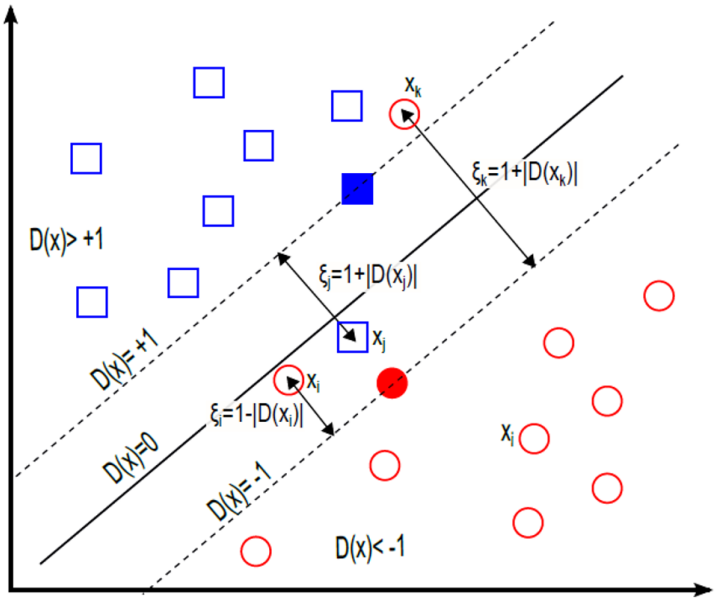

Figura 2.10: Ejemplo de las muestras no-separables, de su variables de holgura, las cuales indican la desviación desde el borde del margen de la clase respectiva. Los ejemplos $x_i, x_j y x_k$ son no-separables ya que sus variables de holgura son mayores que cero, pero mientras que $x_j y x_k$ están mal clasificados al estar en lado incorrecto de la frontera de decisión, x_i esta bien clasificado

En este caso la función que se ha de optimizar tiene que contemplar los errores de clasificación que esta cometiendo el hiperplano de separación:

$$f(w, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \tag{2.71}$$

Donde C es una constante de un valor elegido por el usuario, que ha de ser grande, y permite controlar la influencia de los ejemplos no separables. En función de su valor se tienen dos casos límites:

- C $= \infty$: Ejemplos perfectamente separables, es decir con su variable de holgura con valor 0
- C $= 0$: Se permite que todos os ejemplos estén mal clasificados, es decir con su variable de holgura con valor ∞

El hiperplano que se define, ha de cumplir las condiciones de las ecuaciones 2.72, 2.73 y 2.74, y se denomina *hiperlplano de margen blando*.

$$min \quad \frac{1}{2} < w, w > +C\sum_{i=1}^{n}\xi_i \tag{2.72}$$

$$s.a. \quad y_i(< w, x_i > +b) + \xi_i - 1 \geq 0 \tag{2.73}$$

$$s.a. \quad \xi_i \geq 0 \quad \text{con i=1,...,n} \tag{2.74}$$

Una vez definidas las restricciones del hiperplano se procede a la obtención del mismo, para ello se sigue un procedimiento análogo al caso con separación lineal:

- Obtención de la función Lagrangiana.
- Aplicación de las condiciones KKT.
- Establecimiento de las relaciones entre las variables del problema dual (α, β) y las del problema primal (w, b, ξ).
- Establecimiento de las restricciones de las variables duales.
- Eliminación de las variables primales de la función Lagrangiana para la obtención del problema dual a maximizar, reflejado en las ecuaciones 2.75, 2.76 y 2.77.

$$max \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j < x_i, xj > \tag{2.75}$$

$$s.a. \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.76}$$

$$s.a. \quad 0 \leq \alpha_i \leq C, \quad \text{con i=1,...,n} \tag{2.77}$$

- Obtención de la solución del problema dual, α^*, para poder obtener finalmente el hiperplano de separación óptima, 2.78.

$$D(x) = \sum_{i=1}^{n} \alpha_i^* y_i < x, x_i > +b^* \tag{2.78}$$

- Obtención del parámetro b^*, para lo cual se ha de elegir un ejemplo x_i que tenga asociado un α_i que cumpla la restricción $0 < \alpha_i < C$, de tal forma que se obtendría b^* como se muestra en la ecuación 2.79.

$$b^* = y_i - \sum_{j=1}^{n} \alpha_i^* y_i < x_j, x_i > \quad \forall \alpha_i \text{ tal que } 0 < \alpha_i < C \tag{2.79}$$

2.6.1.3 Clasificadores SVM con separación no lineal

Para este caso, se cuenta con conjuntos de ejemplos no separables linealmente, para lograr su separación no basta con el uso de las técnicas mostradas anteriormente, es necesario definir espacios transformados de alta dimensionalidad y buscar hiperplanos de separación óptimos en tales espacios transformados, a los cuales se les denominara *espacio de características*.

Para ello se define la función de transformación $\Phi : \mathsf{X} \rightarrow F$ que hace corresponder cada vector de entrada x con un punto en el espacio de características,F, donde $\Phi(x) = [\phi_1(x), ..., \phi_m(x)]$ y $\exists \phi_i(x), i = 1, ..., m$, tal que $\phi_i(x)$ es una función no lineal.

Tras ello se define la función de decisión, $D(x)$, como se muestra en la ecuación 2.80, y en su forma dual, ecuación 2.81.

$$D(x) = (w_1 \phi_1(x) + \ldots + w_m \phi_m(x)) =< w, \Phi(x) > \tag{2.80}$$

$$D(x) = sum_{i=1}^{n} \alpha_i^* y_i K(x, x_i) \tag{2.81}$$

Donde $K(x, x^{'})$ se denomina *función kernel*, la cual se define como una función $K : XxX \rightarrow R$ que asigna a cada par de elementos en el espacio de entrada,X un valor real correspondiente al producto escalar de las imágenes de dichos elementos en un nuevo espacio de características(F), ver imagen 2.11, tal y como se muestra en la ecuación 2.82.

$$K(x, x^{'}) =< \Phi(x), \Phi(x') >= (\phi_1(x)\phi_1(x') + \ldots + \phi_m(x)\phi_m(x')) \tag{2.82}$$

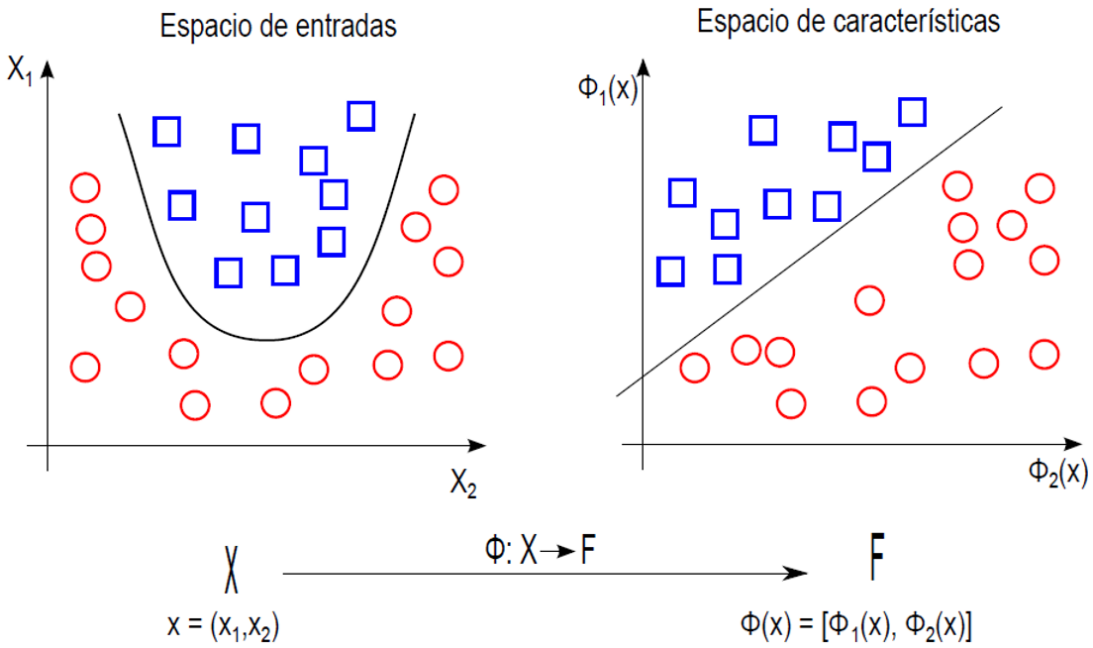

Figura 2.11: Ejemplo de transformación de un espacio de entrada inicial hasta el espacio de características, en el cual se busca la decisión lineal

En la actualidad existen multitud de *kernel* para poder emplear en estos casos, a continuación se muestran los mas conocidos.

- Lineal:

$$K(x, z) = x^T z \tag{2.83}$$

- Cuadrático:

$$K(x, z) = (x^T z)^2 \tag{2.84}$$

- Polinomial de grado d:

$$K(x, z) = (x^T z)^d \tag{2.85}$$

- RBF(Radial Basis Funcion):

$$K(x, z) = exp(-\gamma \|x - z\|^2) \tag{2.86}$$

2.6.2 Analisis de las componentes principales

La técnica del análisis de las componentes principales, *PCA*, fue desarrollada por Jolliffe en 1986, [14] y [15], y consiste en la reducción de la dimensionalidad. La aplicación de la técnica de *PCA* tiene sentido

en el caso de que exista una alta correlación entre las variables de entrada, ya que en el caso de existir tal alta correlación puede existir información redundante, y por tanto se puede reducir el número de variables de entrada que muestren la variabilidad de los datos.

La técnica de análisis de las componentes principales (PCA) consiste en la obtención de un espacio de dimensionalidad menor al espacio de datos de entrada, analizando principalmente la variabilidad del conjunto de datos de entrada.

En un caso general en el que se tengan n datos de entrada, los cuales tengan cada uno de ellos m variables, podemos buscar el número de variables p, con $p < m$, con los cuales podemos aproximar cada uno de los datos de entrada. Con ello hemos reducido de dimensión n a dimensión p, procurando siempre que la variables seleccionadas estén incorreladas y recojan la mayor parte de información posible, por lo que se disminuye la complejidad de la resolución.

Existen dos formas de aplicar la técnica PCA:

2.6.2.1 Metodo basado en correlaciones

Para aplicar la técnica PCA con el método de las correlaciones se parte de una matriz compuesta por los elementos de la matriz de correlaciones, considerando el valor de todas m componentes de los n datos de entrada. Dado que la matriz de correlaciones es simétrica, tiene la característica implícita de ser diagonalizable, y los **m** valores de la diagonal principal forman las m componentes principales. Ha de cumplirse en todo momento que el sumatorio de las componentes principales sea igual a 1.

Una vez obtenidas las componentes principales se ordenan en función de su peso, obteniendo una relación como la observada en el gráfico 2.12, empleado en [16]

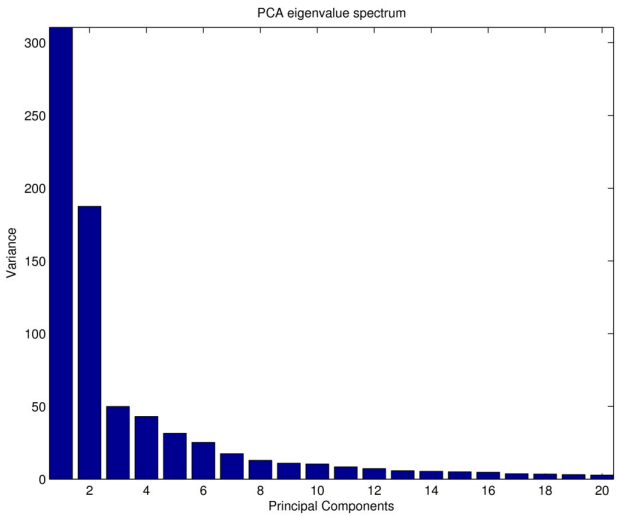

Figura 2.12: Diagrama de las 2 fases principales del proceso de detección y conteo de personas.

En el gráfico anterior se puede observar como las dos primeras componentes principales representan en gran medida la totalidad de las muestras, aunque con esta reducción de dimensionalidad se está perdiendo una pequeña parte de información, el método PCA basado en correlaciones tiene unos grandes resultados, ya que cada una de las variables analizadas puede ser representadas como combinación lineal de las componentes principales.

2.6.2.2 Metodo basado en covarianzas

Al igual que el método anterior, en este caso se parte de n datos con m características cada uno de ellos, y se pretende reducir la dimensionalidad de cada uno de los datos de m a p, con $m < p$, dando lugar a la menor perdida de información posible.

En este caso para formar la matriz con la que se trabajará, se centran los datos de entrada en una matriz de media 0, para lo que se les resta la media de cada columna, y posteriormente se procede a su autoescalamiento, es decir una vez que estén centrados a media 0 se divide cada columna por su desviación estándar. Con ello obtenemos la siguiente matriz X de la ecuación 2.87.

$$X = \sum_{i=1}^{p} t_i p_i^T + E \tag{2.87}$$

Donde t_i son vectores conocidos como *scores*, los cuales contienen la información acerca de la relación de unas muestras con las otras, y son ortogonales. Los vectores p_i aportan información de la relación existente entre las diferentes variables, y se denominan *loadings*. Al seleccionar menos componentes principales que variables que contiene cada dato, $p < m$, se produce un error acumulado en la matriz E.

Tras ello se procede a la descomposición de la matriz de covarianza en los vectores propios de ella, como se observa en las ecuaciones 2.88, 2.89 y 2.90.

$$cov(X) = \frac{X^T X}{n-1} \tag{2.88}$$

$$cov(X)p_i = \lambda_i p_i \quad \text{i=1,...,m} \tag{2.89}$$

$$\sum_{i=1}^{m} \lambda_i = 1 \tag{2.90}$$

Donde el autovalor λ_i esta asociado al autovector p_i. Es decir los autovalores contienen la cantidad varianza de la información de cada una de las componentes principales. Obteniendo una disminución de la misma en función del decremento del índice de las componentes principales, siendo la primera componente principal la más importante.

A continuación se va a mostrar el proceso seguido por la técnica PCA, como se comenta en [17]. La técnica PCA habitualmente es empleada para disminuir la dimensionalidad de vectores de características en visión artificial, pero tras esta etapa puede ser empleada como clasificador, deduciendo la clase a la que pertenece dicho objeto a partir del conocimiento a priori de cada clase.

La clasificación aplicando la técnica PCA tiene 2 etapas, off-line y on-line:

- Proceso off-line: se parte de un conjunto de datos de entrenamiento, compuesto por N vectores de medida, cuya pertenencia a las diferentes clases que se van a evaluar es conocida. Tras ello para cada clase se sigue cualquiera de los métodos comentado anteriormente para obtener los autovalores y autovectores asociados. A continuación se procede a seleccionar el número de componentes principales que se emplearan para cada clase, y con los autovectores asociados a los autovalores de mayor valor, comprendidos dentro del número de componentes principales, se procede a formar la matriz de transformación $U = [u_1, ..., u_m]_{nxm}$, donde u_i con $0 \leq i \leq m$ son los autovectores seleccionados para cada una de las clases. Por ello las columnas de la matriz son linealmente independientes y forman un espacio ortonormal.

- Proceso on-line: se parte de los datos de la etapa off-line y un vector de medidas de entrada, y, se comienza transformando el vector de entrada a los espacios de características denotados para cada clase por su matriz de transformación correspondiente, como se muestra en la ecuación 2.91.

$$\Omega = U^T \Phi = U^T (y - \Psi) \tag{2.91}$$

  Donde $\Phi \in \Re^n$ es el vector de medida con media nula sobre el que se realiza la transformación, $\Omega \in \Re^m$ es el vector resultado de la transformación, $U \in \Re^{nxm}$ es la matriz de transformación que ha sido previamente obtenida en la fase offline, y Ψ es el vector de media de los vectores de medida empleados en la obtención de la matriz U.

  Tal que una vez obtenida la transformación al espacio de características, se puede recuperar el vector Φ, mediante la ecuación 2.92, pero el vector recuperado no seria igual que ϕ, sino que al recuperarlo se estaría incorporando un error o distancia de recuperación, ε_{PCA}, en el vector recuperado $\hat{\Phi}$, la ecuación 2.93 la obtención del error de recuperación.

$$\hat{\Phi} = U\Omega \tag{2.92}$$

$$\varepsilon_{PCA} = \|\Phi - \hat{\Phi}\| \tag{2.93}$$

Este error será menor cuanto mayor sea la similitud entre el objeto evaluado y los empleados en la etapa de entrenamiento (*offline*), permitiendo determinar si un vector pertenece a una clase dada, o no.

2.7 Sistemas de seguimiento

En los sistemas actuales nos encontramos ante la necesidad de tener información sobre las variables en situaciones desconocidas, por ejemplo predecir el valor de una variable dentro de un rango de error. Para ello se ha hecho uso de observadores y predictores de estados, tales como el observador de Luenberger, [18], que otorga una estimación de estado interno no mesurable de un sistema dinámico lineal a partir de las mediciones realizadas a la entrada y salida del sistema. La predicción del valor que tendrá de una variable aplicado a la visión es de gran interés, ya que podemos predecir la trayectoria, velocidad, forma o área de un objeto que se encuentre dentro de la región de interés estudiada.

En la actualidad se trabaja con sistemas de seguimiento que tengan en cuenta multitud de factores, por ello el ruido que comúnmente tenemos en los sistemas no ha de resultar un inconveniente, gracias a sistemas como el filtro de kalman o el filtro de partículas podemos evaluar estos sistemas con adicción de ruido.

A continuación se estudiaran el Filtro de Kalman y el Filtro de partículas así como sus derivados.

2.7.1 Filtro de Kalman

El Filtro de Kalman fue desarrollado por Rudolf E. Kalman en 1960, entre sus numerosas ventajas destaca la posibilidad de obtener datos de entrada con ruido, algo muy común en cualquier sistema real. El Filtro de Kalman es un estimador del valor de un vector de estado a partir de los valores de las observaciones que se realizan, considerando estas últimas una incertidumbre. Este sistema de seguimiento se compone de un algoritmo recursivo por lo que puede ser empleado en tiempo real haciendo uso de las variables actualmente observadas, el estado calculado previamente y de su matriz de incertidumbre. En la imagen 2.13 se observan las etapas del Filtro de Kalman.

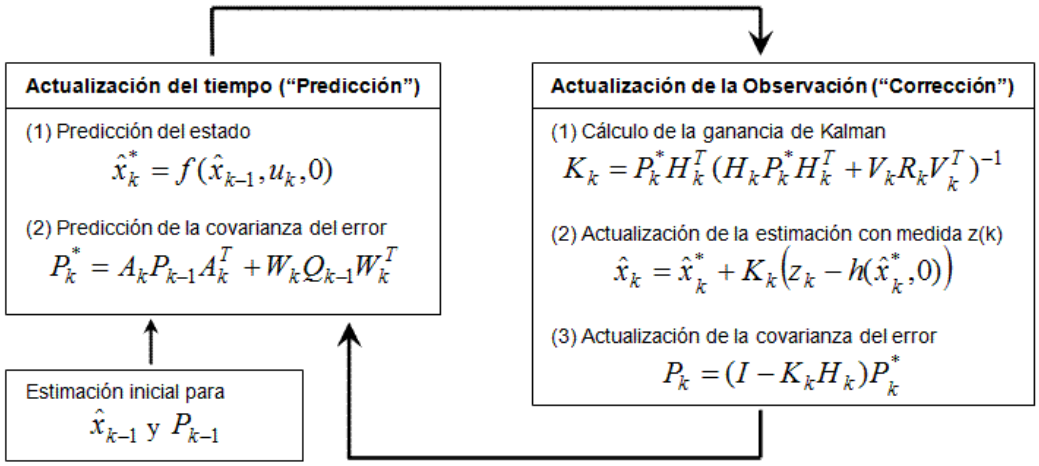

Figura 2.13: Diagrama de las etapas del Filtro de Kalman extraído de [3].

El Filtro de Kalman puede ser aplicado tanto a sistemas continuos como discretos, y consta principalmente de 2 etapas que se comentaran a continuación: predicción y corrección. En [19] se describen las etapas que se comentan a continuación.

2.7.1.1 Predicción

En esta etapa se determina a evolución del vector de estado, $\hat{X}_k^*$, en el tiempo a partir del modelo del sistema, ecuación 2.94, y del vector de estado en la iteración anterior, $\hat{X_{k-1}^*}$. El sistema viene definido por la matriz A, la cual determina un modelo de velocidad constante, donde el parámetro T indica el tiempo transcurrido entre dos medidas consecutivas, lo que actualiza el valor del vector de estado en función del estado anterior. La matriz B representa la actualización del vector de estado en función de la entrada, u_k. Y para concluir la matriz W, la cual representa el ruido del proceso, independiente, blanco y con distribución de probabilidad normal, es decir un ruido blanco gaussiano.

$$\hat{X}_k^* = A\hat{X}_{k-1}^* + Bu_k + W \tag{2.94}$$

$$A = \begin{pmatrix} 1 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.95}$$

$$P_k^* = AP_{k-1}A^T + Q \tag{2.96}$$

Ademas se calcula la proyección de la covarianza del error P_k^*, conocido como covarianza de error a priori, la cual depende de la matriz A y de la matriz Q, la cual representa la covarianza de la perturbación aleatoria del proceso que trata de estimar el estado.

Tras la predicción de la medida se procede a la predicción de la observación, la cual viene descrita a través de la matriz H que relaciona el estado, con la medida y con la matriz V que representa el ruido de la medida, el cual es un ruido blanco gaussiano.

2.7.1.2 Corrección

Una vez obtenidas las predicciones del vector de estados y de la observación, se determina el valor de la constante de Kalman K_t, ver ecuación 2.97, para ello se recurre a la covarianza de error a priori P_k^*, la matriz H mencionada anteriormente, y a la matriz R que representa la covarianza de la pertubación aleatoria de la medida.

$$K_k = P_k^* H^t (H P_k^* H^T + R)^{-1} \tag{2.97}$$

Una vez obtenido el valor de la constante de Kalman, K_k se procede a calcular el nuevo vector de estados estimado, ecuación 2.98.

$$\hat{X}_k = \hat{X}_k^* + X_k(Z_k + H\hat{X}_k^*) \tag{2.98}$$

Y por último se actualiza la covarianza del error a posteirori, ecuación 2.99.

$$P_k = (I - K_k H) P_k^* \tag{2.99}$$

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.100}$$

A continuación se muestra un gráfico que explica brevemente las diferentes etapas del Filtro de Kalman

[hacer la imagen de kalman, poner en google filtro de kalman etapas y aparece]

2.7.2 Filtro de partículas extendido

El Filtro de Kalman que se presentó anteriormente no presenta carácter multimodal, es decir un filtro únicamente puede ser empleado para realizar una predicción. En el caso de querer aplicar el filtro de Kalman al sistema de detección de personas que se propone en este TFG, se debería implementar un Filtro de kalman por cada individuo a seguir dentro del entorno, algo que en el caso de tener numerosos individuos puede no resultar recomendable por los posibles errores que proporciona. Por ello se presenta a continuación el filtro de partículas extendido, un sistema con carácter multimodal.

El Filtro de Partículas Extendido se basa en la combinación de métodos probabilísticos de estimación con métodos determinísticos de asociación.

El algoritmo probabilístico utilizado como base es un filtro de partículas, que gracias a su carácter multimodal permite modelar con una única función la posición de multitud de items en cada instante. El filtro de partículas básico consta de 3 etapas, predicción, asociación+corrección y selección. Pero dado que el sistema de detección implementado en este TFG incluye la determinación del número de personas dentro del entorno, es necesario añadir la etapa de reinicializacion en el modelo básico, lo que se conoce como filtro de partículas extendido. Como principal ventaja, el filtro se robustece, lo que junto con la adicción de un nuevo clasificador implementado a la salida del filtro, se obtendrá un filtro de partículas extendido con proceso de clasificación (*XPFCP*)

A continuación se muestra el diagrama 2.14, que indica las diferentes etapas del *XPFCP*.

El filtro de partículas básico, *PF*, es un estimador recursivo probabilístico cuyo funcionamiento esta basado en la representación discreta de la creencia, para lo que se emplea un conjunto S_t de n_t muestras denominadas partículas($S_t = \{s_{ti} = x_t^i\}_{i=1}^{n_p}$),que son copias del vector de estado x_t del sistema a estimar, cada una de las muestras está a su vez ponderada por un peso normalizado $w_t = \{\tilde{w}_t^i\}_{i=1}^{n_p}$. El cual

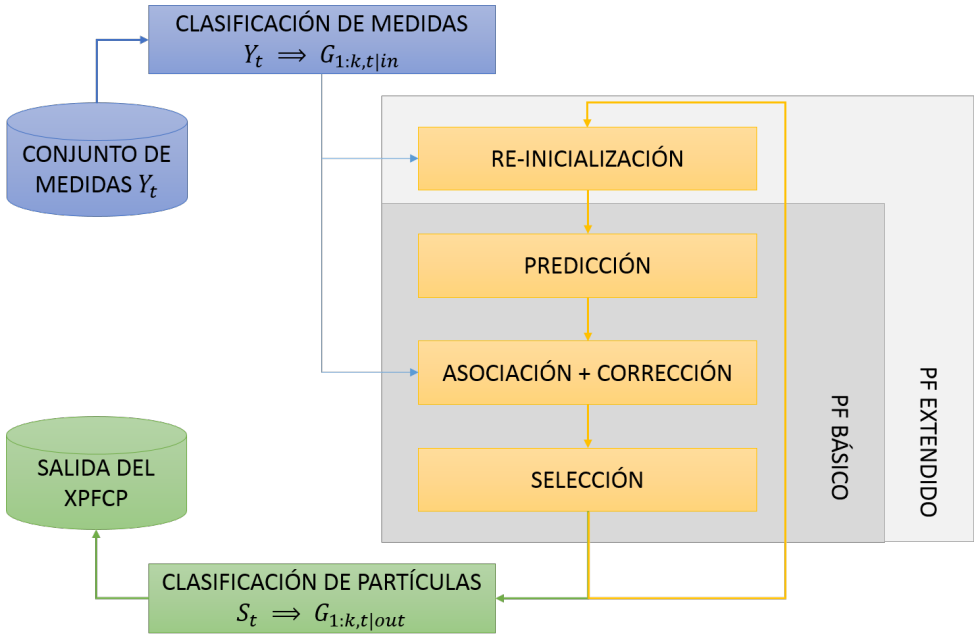

Figura 2.14: Diagrama de las 2 fases principales del proceso de detección y conteo de personas.

caracteriza su probabilidad de certidumbre dentro de la distribución de probabilidad global. A través de ello se obtiene la salida del filtro de partículas con el valor final de la estimación, ver ecuación 2.101

$$p(x^t_{1:n_p}|y_{1:t}) \cong S_t = \{x^i_t, \tilde{w}^i_t\}^{n_p}_{i=1} \tag{2.101}$$

El Filtro de Partículas permite su implementación en tiempo real sin discretizar el espacio de estado, o linealizar el modelo de sistema, por lo que permite obtener resultados de estimación mas precisos que el Filtro de Kalman.

A continuación se van a describir brevemente las etapas que forman el filtro de partículas.

2.7.2.1 Reinicialización

La adición de esta etapa permite la incorporación de nuevas hipótesis de seguimiento al filtro de partículas, esto se realiza mediante la incorporación de un nuevo conjunto de partículas al conjunto, $S_{t-1} \cong p(x_{t-1}|y_{1:t-1})$, lo que caracteriza la probabilidad a posteriori obtenida en el instante de tiempo anterior. Lo que da lugar a la generación de un nuevo conjunto de partículas,$\hat{S}_{t-1}$.

Las partículas se obtienen a partir del resultado generado en la clasificación de medidas en ese instante temporal ($Y_{t-1} \Rightarrow G_{1:k,t-1|in}$), gracias a lo cual la nueva función de densidad de probabilidad, $\hat{p}(x_{t-1}|y_{t-1})$,representa de forma robusta las diferentes hipótesis y evita que se empobrezca el conjunto de partículas empleadas.

El número de partículas del filtro, n_p, ha de mantenerse constante, por lo que el número de partículas que se añaden en cada etapa ha de ser igual al número de partículas eliminadas en la etapa de selección, $n_{m,t-1}$. La relación entre el número de partículas eliminadas en la etapa de selección y el número de partículas totales del filtro se denomina $\gamma_{p,t}$, y con ella puede expresarse la función densidad de probabilidad del filtro de partículas, ecuación 2.103

$$\gamma_{p,t} = \frac{n_{m,t-1}}{n_p} \tag{2.102}$$

$$\hat{p}(x_{t-1}|y_{t-1}) = \gamma_{p,t} p(x_{t-1}|y_{1:t-1}) + (1-\gamma_{p,t}) p(G_{1:k,t-1|in}) \tag{2.103}$$

2.7.2.2 Predicción

Durante la etapa de predicción del filtro de partículas. todas las partículas que proceden del paso de reinicialización, $\hat{S}_{t-1} = \{\hat{x}^i_{t-1}, \frac{1}{n}\}_{i=1}^{n_p} \cong \hat{p}(x_{t-1}^{1:n_p}|y_{1:t-1})$, se propaga al siguiente instante de tiempo mediante el modelo de estado $p(x_t|x_{t-1})$, como muestra la ecuación 2.104. Gracias ha esta propagación de las partículas se obtiene el valor a priori de la creencia , $S_{t|t-1}$, el cual viene descrito por un conjunto de muestras y sus correspondientes pesos.

$$x_{t|t-1}^{1:n_p} = x_{t-1}^{1:n_p} p(x_t|x_{t-1}) \tag{2.104}$$

$$S_{t|t-1} = \{x^i_{t|t-1}, \tilde{w}_{t-1}\}_{i=1}^{n_p} \cong \hat{p}(x_t^{1:n_p}|y_{1:t-1}) \tag{2.105}$$

2.7.2.3 Corrección

En el paso de corrección se obtiene la función de probabilidad a posteriori, $p(x_t^{1:n_p}|y_{1:t})$, a partir de la creencia muestreada, $p(x_t^{1:n_p}|y_{1:t-1})$, calculada en la etapa de predicción, la cual se obtiene mediante el muestreo de Monte-Carlo, ecuación 2.106.

$$p(x_t^{1:n_p}|y_{1:t-1}) = \sum_{i=1}^{n_p} \tilde{w}_t^i \delta(x_t^i) \tag{2.106}$$

Los pesos normalizados de las muestras, $w_t = \{w_t^i\}_{i=1}^{n_p}$, caracterizan la creencia, $p(x_t|y_{1:t})$, y su obtención se realiza de forma recursiva como se muestra en las ecuaciones 2.107 y 2.108.

$$w(x_{0:t}) = w(x_{0:t-1}) \frac{p(y_t|x_t)p(x_t|x_{t-1})}{q(xt|x_{0:t-1}, y_{1:t})} \tag{2.107}$$

$$\tilde{w}_t^i = \frac{w_t^i}{\sum_{i=1}^{n_p} w_t^i} \tag{2.108}$$

Donde $q(xt|x_{0:t-1}, y_{1:t})$ es la función de aproximación a la creencia, la cual puede sustituirse por el modelo de actuación del sistema para simplificar la ecuación,2.109.

$$w(x_{0:t}) = w(x_{0:t-1}) p(y_t|x_t) \Rightarrow w_t^{1:n_p} = w_{t-1}^{1:n_p} p(y_t|x_t^{1:n_p}) \tag{2.109}$$

En el caso del XPFCP que se propone en [la tesis de Marta Marrón] la función de verosimilitud considerada en la etapa de corrección se genera a partir de los centroides $g_{1:k,t|in}$ obtenidos al final de la salida del clasificador de medidas. Estos centroides representan de forma robusta al conjunto de medidas, como se muestra en la ecuación 2.110.

$$w_t^{1:n_p} = w_{t-1}^{1:n_p} \cdot p(g_{1:k,t|in}|x_t) \tag{2.110}$$

El valor discreto de cada peso $\tilde{w}_t^i$ se obtiene buscando el máximo de la gaussiana entre las realizaciones que se obtienen de la misma para cada una de las medidas. Tal máximo viene dado por la distancia mínima entre la media de la gaussiana y la medida más próxima a ésta, como muestra la ecuación 2.111.

$$w_t^i = w_{t-1}^i \cdot e^{-\frac{d^2_{min,i,t}}{2 \cdot O}} \tag{2.111}$$

Donde $d_{min,i,t}$ representa el valor mínimo de la distancia entre cada proyección de la partícula x_t^i en el espacio de medidas y cada una de las medidas $\{y_t^j\}_{j=1}^{m_t}$ en el instante t, siendo O la varianza de la gaussiana que coincide con la covarianza del modelo de observación.

Tras la obtención de todos los pesos de las partículas se procede a la normalización de los mismos.

Por ello a la salida de las etapas de predicción y corrección se obtiene el conjunto de partículas $S_t = \{x_t^i, \tilde{w}_t^i\}_{i=1}^{n_p}$ que describe la creencia $p(x_t|y_{1:t})$. El objetivo de la etapa de corrección es la obtención de un peso $\tilde{w}_t^i$ para cada partícula x_t^i que pondere la probabilidad de acierto de dicha partícula en el proceso de estimación.

2.7.2.4 Selección

La etapa de selección se ejecuta tras los pasos de predicción y corrección, y en ella se seleccionan del conjunto de muestras ponderadas de la creencia $S_t' = \{x_t^i, \tilde{w}_t^i\}_{i=1}^{n_p}$ un conjunto de $n_p - n_{m,t}$ partículas 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.

2.7.2.5 Modelo del sistema

En el sistema implementado en este trabajo se plantea un modelo discreto, genérico y de primer orden, generado a partir de la posición de las diferentes personas que se desean evaluar dentro del entorno en cada instante. Mediante las siguientes ecuaciones se modela el sistema.

$$\begin{aligned} x_{t+1} &= x_t + v_{x,t} \cdot t_s \\ y_{t+1} &= y_t + v_{y,t} \cdot t_s \\ z_{t+1} &= z_t \end{aligned} \tag{2.112}$$

Se definen los vectores de estado x_t y de medida y_t en el instante temporal t mediante los vectores de dimension 5x1 que se muestran en las ecuaciones 2.113 y 2.114.

$$x_t = [X_{w,t} \quad Y_{w,t} \quad Z_{w,t} \quad V_{x,t}^w \quad V_{y,t}^w]^T \tag{2.113}$$

$$y_t = [X_{w,t} \quad Y_{w,t} \quad Z_{w,t} \quad V_{x,t}^w \quad V_{y,t}^w]^T \tag{2.114}$$

Donde $X_{w,t}$ representa la posición en x en el instante t, $Y_{w,t}$ representa la posición en y en el instante t, $Z_{w,t}$ representa la posición en z en el instante t, $V_{x,t}^w$ representa la velocidad en x en el instante t y $V_{y,t}^w$ representa la velocidad en y en el instante t.

Las ecuaciones que modelan el sistema se pueden expresar en variables de estado como se muestra en la ecuación 2.115.

$$\begin{aligned} x_{t+1} &= A \cdot x_t \\ y_t &= C \cdot x_t \end{aligned} \tag{2.115}$$

Donde A y C son las matrices representadas en la ecuación 2.116, en las que t_s representa el tiempo de muestreo.

$$A = \begin{pmatrix} 1 & 0 & 0 & t_s & 0 \\ 0 & 1 & 0 & 0 & t_s \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad y \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.116}$$

2.8 Conclusiones

En esta sección se han comentado todos los fundamentos teóricos para la obtención de un sistema de detección y conteo de personas basándose en imágenes de profundidad. Se comenzó tratando los fundamentos de las cámaras de tiempo de vuelo, así como los errores mas comunes que se pueden encontrar en las imágenes adquiridas, y gracias a ello poder eliminarlos durante el desarrollo del sistema. Se analizó la base teórica de los extractores de características para imágenes en profundidad, así como los sistemas de clasificación que se emplearan en el sistema. Finalmente se profundizó en los sistemas de clasificación de carácter tanto unimodal, Filtro de Kalman, como multimodal, Filtro de Partículas Extendido. Todos los fundamentos tratados en esta sección permitirán la implementación del sistema, el cual se tratara en el capítulo 3 .

Capítulo 3

Implementación del sistema de detección y conteo de personas basado en imágenes de profundidad

3.1 Introducción

El presente capítulo se centrará en las etapas desarrolladas para la obtención de el sistema de detección y conteo de personas implementado en este Trabajo Fin de Grado. Los Sistemas de detección y conteo de personas constan de diferentes etapas, las cuales dependen en gran medida entre sí, por lo que la determinación de las características y especificaciones de cada una de ellas resulta determinante para el resultado final.

A continuación se describen en detalle las diferentes etapas por las que se transcurre para la obtención del sistema. También se especifican los pasos que se realizan para la obtención de un sistema robusto y fiable. Para facilitar la lectura se introduce de nuevo el diagrama de bloques del sistema, figura 3.1.

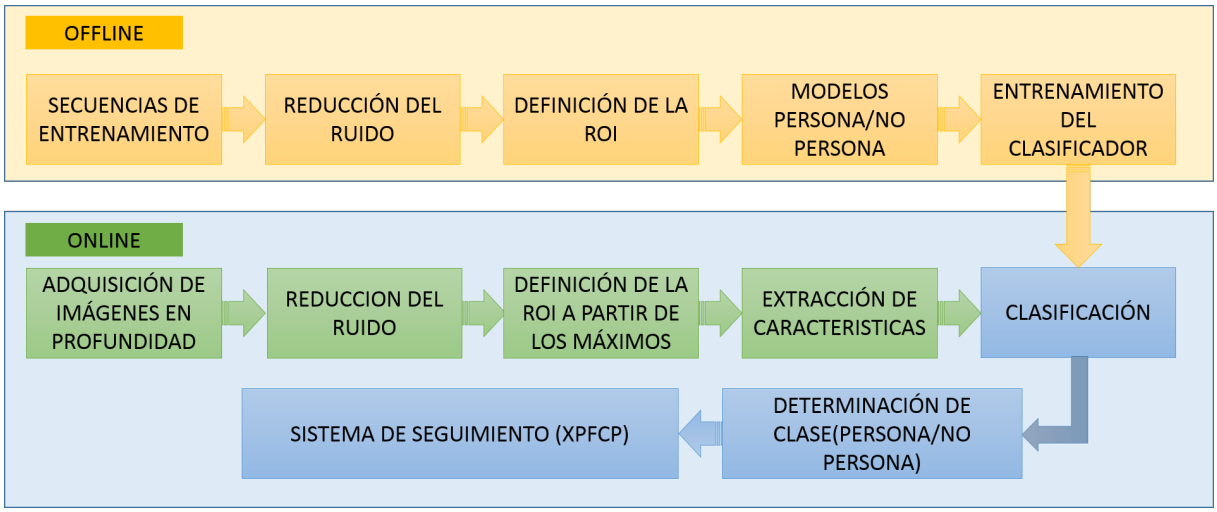

Figura 3.1: Diagrama de las 2 fases principales del proceso de detección y conteo de personas.

- **Sistema de adquisición de imágenes:** el sistema se compone de dos fases, una fase offline y otra online, en un primer lugar se obtiene un conjunto de muestras de imágenes que se emplean

como base de datos para el entrenamiento del sistema. La base de datos elegida es ampliamente representativa de las posibilidades que pueden darse en el entorno, ya que de ello depende que en la fase online se contemplen todas las posibilidades. El sistema de adquisición de imágenes en el proceso online es similar al del proceso offline pero en su caso las imágenes obtenidas son las del entorno, sin que se de una selección previa, ya que se trata de un sistema de tiempo real.

- **Tratamiento de las imágenes:** en los dos procesos, online y offline, se tratan las imágenes para eliminar la mayor cantidad de ruido e información innecesaria. Este proceso es de suma importancia ya que las cámaras de tiempo de vuelo recogen mucho ruido, sobre todo por la parte periférica de entorno, donde se acumula en gran medida
- **Extracción de características:** una vez obtenidas las imágenes, y procesadas para la eliminación del ruido, se procede a la extracción de características. En función de la posición de la cámara, el tipo de imagen, en color, escala de grises o imágenes de profundidad, se evalúan diferentes descriptores hasta llegar a un descriptor óptimo. Mas adelante se mostrarán en profundidad el proceso seguido hasta la selección del descriptor mas adecuado para este sistema.
- **Sistema de clasificación:** este sistema parte de los vectores de características extraídos de las imágenes, tanto en el proceso de entrenamiento como en la fase online. Con los vectores de características de la fase offline se entrena el sistema, para que en la fase online el clasificador tome la decisión correcta y asocie el vector de características de la imagen a una de las clases previamente declaradas.
- **Sistema de seguimiento:** una vez conocida la pertenencia de los posibles items que se encuentran en el entorno evaluado a las clases declaradas, en el caso de que se trate de una persona se procede a su seguimiento, gracias al cual se conoce si se trata de una persona que ya se encontraba en un *frame* anterior, la variación de su posición conforme a *frames* anteriores, y por tanto su velocidad.

3.2 Sistema de adquisición de imágenes

El sistema de detección y conteo implementado tiene como principal fuente de información las imágenes que aporta la cámara de tiempo de vuelo seleccionada, *Kinect II*. Esta cámara aporta 3 tipos de imágenes: en escala de grises, en RGB y en profundidad, en nuestro caso se selecciona la información en profundidad, debido a que es la mas adecuada para la detección y no identificación de las personas que se encuentren en el entorno. Este detalle resulta determinante, ya que la protección de la intimidad de las posibles personas que se encuentran en el entorno evaluado es básica. Por ello se ha determinado que la cámara se coloca en posición cenital para evitar oclusiones y no obtener imágenes de los rostros de las personas que puedan servir para su identificación.

Una vez obtenidas las imágenes, tanto para la fase de entrenamiento, como para su procesamiento para la fase online se continua con la etapa de pre-procesado. Ésto se debe a que la cámara empleada Kinect II como la gran mayoría de las cámaras de tiempo de vuelo proporciona imágenes con gran cantidad de ruido.

3.3 Pre-procesado de las imágenes de entrada

Como se comentó en la sección anterior, las imágenes obtenidas de la cámara de tiempo de vuelo que se emplea en este Trabajo Final de Grado, la Kinect II, contienen gran cantidad de ruido, especialmente en la periferia, por ello se reduce en la medida de lo posible.

A continuación se observa en la imagen 3.2 una trama sin tratar su ruido, en ella se pueden apreciar principalmente tres ruidos o medidas erróneas:

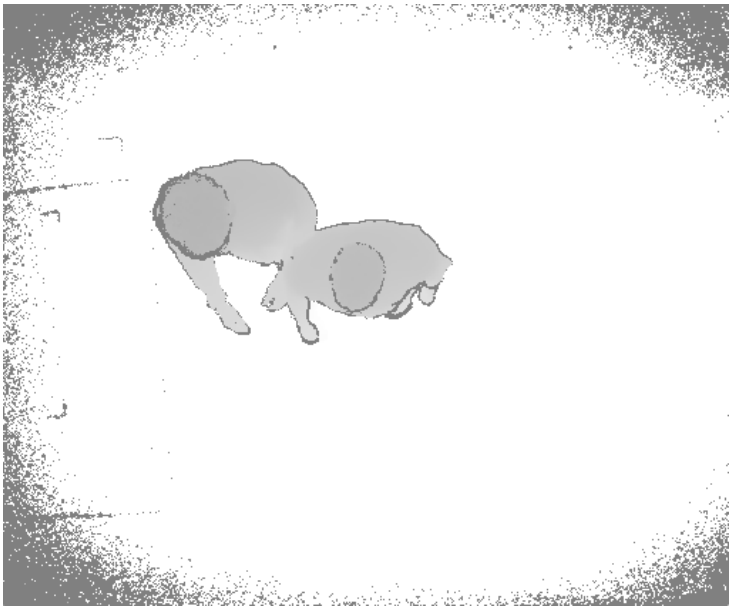

Figura 3.2: Imagen obtenida de un *frame* del entorno evaluado sin tratamiento del ruido.

- El ruido periférico: este ruido se debe a que cuanto mayor es la distancia a la que se encuentra el objeto, siempre que se encuentre dentro de la distancia máxima que se puede medir sin error, menor es la amplitud de la señal recibida medidas dentro de la distancia máxima, al estar alejados, la señal recibida es de menor amplitud, por lo que se producen mas medidas erróneas. Las partes del entorno que se encuentran en la periferia del campo de visión se encuentran a mayor distancia, por lo que la posibilidad de encontrar medidas erróneas crece notablemente.

- Ruido debido a medidas erróneas: en el perímetro de la persona que se muestra en la imagen se puede observar un conjunto de medidas erróneas, estas medidas se producen al existir una gran diferencia entre las diferentes medidas recogidas durante el tiempo de integración. Además, si la persona u objeto se está moviendo, pueden aparecer errores debidos a los artefactos en movimiento.

- Ruido producido por la tonalidad: debido a la tonalidad de los objetos o personas presentes en el entorno se puede dar lugar a una mayor cantidad de ruido. Esto se debe a que cualquier elemento oscuro absorbe una mayor cantidad de luz, por los que la señal reflejada tiene una menor amplitud.

En los casos en que se detecta que una medida de distancia es errónea, la Kinect II utilizada en este trabajo marca ese píxel asociándole un valor de distancia 0.

Para reducir la cantidad de ruido en la imagen se han realizado 2 procesos, los cuales se presentan a continuación.

En primer lugar se eliminan los píxeles erróneos o nulos que se encuentran en la imagen, para ello por cada uno de estos píxeles, se realiza una búsqueda de píxeles con valores válidos en su entorno de vecindad 2. En caso de que se encuentren píxeles válidos en dicho entorno, el píxel erróneo se corrige asignándole el valor medio de todos su píxeles vecinos válidos. Se ha considerado un nivel 2 de vecindad debido a que se asume que a esa distancia las medidas deben corresponder al mismo elemento, encontrándose fuertemente correladas entre sí. Gracias a ello se reduce ámpliamente el ruido que se puede encontrar en la imagen.

Por último, dado que en la periferia se sigue encontrando ruido, se procede a umbralizar parte de los píxeles. En este sistema la cámara de profundidad se encuentra colocada en posición cenital, y la información relevante de las posibles personas que se encuentran en el entorno, se encuentra en la franja comprendida desde el máximo de su cabeza hasta los hombros, ya que la zona que se encuentra por debajo de esta franja se encuentra oclusionada, pero en el caso de este sistema no es relevante. Por ello una vez conocido el máximo valor valido representativo de una altura dentro del entorno, se asocia un valor de umbralización a todas las medidas que se encuentren por debajo de la posible zona donde se pueda encontrar la franja de información de las personas del entorno.

Una vez realizado estos dos procesos se realiza una última fase, en la cual se asocia el mismo valor de umbralización empleado anteriormente, a los píxeles que se encuentran en la periferia del entorno determinado por un recuadro preestablecido. En la imagen 3.3 se puede observar el resultado final de la reducción de ruido.

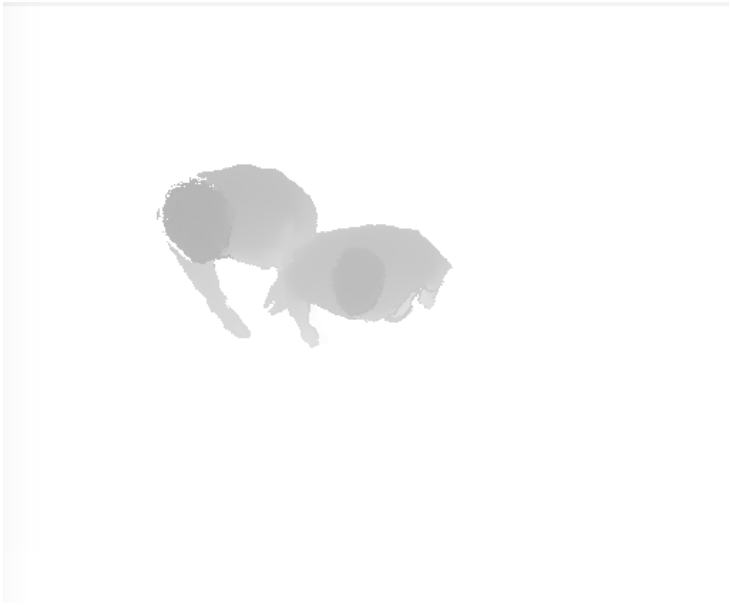

Figura 3.3: Imagen obtenida de un *frame* del entorno evaluado tras el tratamiento del ruido.

3.4 Obtención de la región de interés

Una vez pre-procesadas las imágenes para la eliminación del ruido, se procede a la detección de máximos. Los máximos detectados permiten obtener regiones de interés a su alrededor, el algoritmo implementado para la detección de los máximos y obtención de las ROIs se explica en [5].

Partiendo de los máximos detectados, se define la ROI como la región de interés perteneciente a un máximo de la cual se extraerán las características. Como se comento en capítulos anteriores la ROI ha de contener información de la cabeza y hombros de las personas evaluadas, lo que supone que para la obtención de la ROI se consideran las medidas cercanas que se encuentren $40cm$ por debajo del máximo detectado.

A fin de obtener la ROI se evalúan los píxeles que pertenecen a ella, para lo cual se establece un radio de vecindad asociado, L, de N niveles de vecindad y 8 direcciones. Para obtener todas las subregiones (SR) que pertenecen a item a analizar se evalúan todas las direcciones. En la imagen 3.4 se muestran los diferentes niveles de vecindad, así como las 8 posibles direcciones.

Con objetivo de que una SR pertenezca a la ROI, se tienen que cumplir las siguientes restricciones:

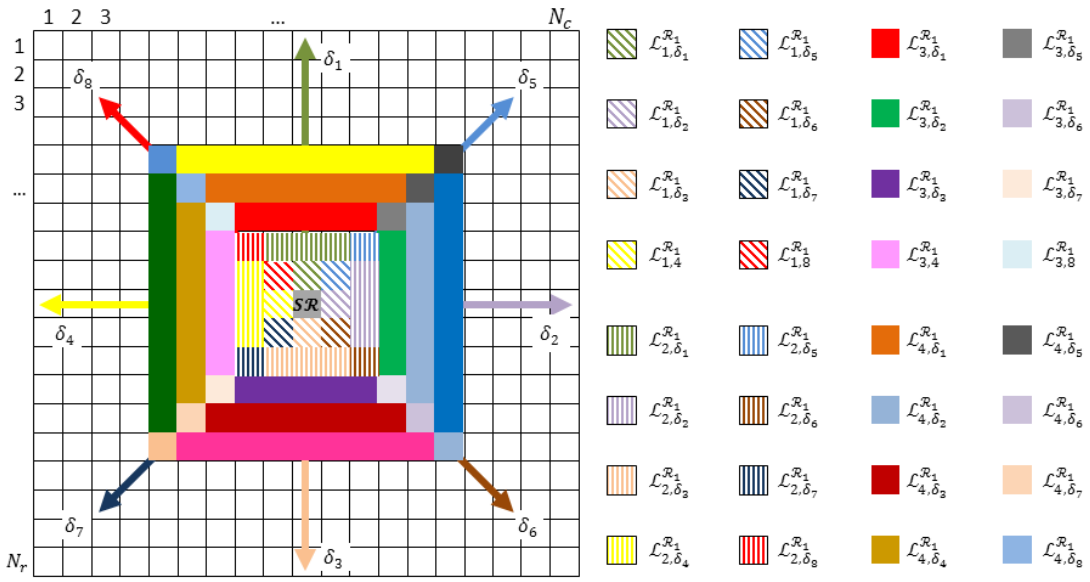

Figura 3.4: Niveles de vecindad y direcciones de búsqueda de las subregiones pertenecientes a la ROI.

- El núcleo de la ROI, denominado $SR^k_{r,c}$, es la SR en la cual se ha localizado el máximo $P^k_{r,c}$, y esta SR siempre pertenece a la $ROI^k_{r,c}$. Donde r y c representan las coordenadas pixélicas de la ROI evaluada dentro del plano imagen, k es un indice que recorre los máximos de la imagen.

- El nivel de vecindad 1, L=1, se compone de las 8 SR vecinas de la $SR^k_{r,c}$. Las SR pertenecientes al nivel de vecindad L=1 que cumplan la condición de la ecuación 3.1 se consideran pertenecientes a la $ROI^k_{r,c}$.

$$h^{max}_{SR} \geq h^{maxSR}_{r,c} - h^{interest} \tag{3.1}$$

  Donde $h^{interest} = 40cm$

- Tras ello se analizan las SR pertenecientes a las direcciones 1, 2, 3 y 4, partiendo desde $P^k_{r,c}$. Se analizan las SR pertenecientes a los niveles de vecindad L=2, 3 y 4 con dirección 1, 2, 3 o 4, esto se debe a que la posible persona no abarcara mas niveles debido a las características de la posición de la cámara en el entorno. Para que una de estas SR pertenezca a $ROI^k_{r,c}$ se han de cumplir las siguientes 4 restricciones:

  - Para que una SR perteneciente a un nivel L=2, 3 y 4 en una dirección 1, 2, 3 o 4 pertenezca a $ROI^k_{r,c}$, es necesario que en el nivel inferior contenga al menos L-1 subregiones en la misma dirección.

  - Para que una SR perteneciente a un nivel L=2, 3 y 4 en una dirección 1, 2, 3 o 4 pertenezca a $ROI^k_{r,c}$, es necesario que la SR anterior en esa misma dirección pertenezca al máximo.

  - El valor del máximo situado en las SR pertenecientes a un nivel L=2, 3 y 4 en una dirección 1, 2, 3 o 4, ha de cumplir la condición de la ecuación 3.1.

  - Para poder diferenciar las SR pertenecientes a un individuo de las de otro en el caso de encontrarse muy juntos, ha de cumplirse la ecuación 3.3 para las SR que se encuentran en una determinada dirección.

$$h^{max}_{SR-1} \geq h^{max}_{SR} \geq h^{max}_{SR+1} \tag{3.2}$$

En el caso de no cumplirse esta condición supone que se encuentra un mínimo entre dos máximos, lo que se interpreta como el limite de las SR pertenecientes a $ROI^k_{r,c}$ en esa dirección.

- Tras ello se analizan las SR pertenecientes a las direcciones 5, 6, 7 y 8, partiendo desde $P^k_{r,c}$, y que se encuentren en los niveles de vecindad L=2, 3 y 4. Para que una de estas SR pertenezca a $ROI^k_{r,c}$ se han de cumplir las siguientes 3 restricciones:
  - Para que una SR perteneciente a un nivel L=2, 3 y 4 en una dirección 5, 6, 7 o 8 pertenezca a $ROI^k_{r,c}$, es necesario que la SR anterior en esa misma dirección pertenezca al máximo.
  - El valor del máximo situado en las SR pertenecientes a un nivel L=2, 3 y 4 en una dirección 1, 2, 3 o 4, ha de cumplir la condición de la ecuación 3.1.
  - Para poder diferenciar las SR pertenecientes a un individuo de las de otro en el caso de encontrarse muy juntos, ha de cumplirse la ecuación 3.3 para las SR que se encuentran en una determinada dirección.

    $$h^{max}_{SR-1} \geq h^{max}_{SR} \geq h^{max}_{SR+1} \tag{3.3}$$

    En el caso de no cumplirse esta condición supone que se encuentra un mínimo entre dos máximos, lo que se interpreta como el limite de las SR pertenecientes a $ROI^k_{r,c}$ en esa dirección.

A partir de las condiciones expuestas anteriormente se obtienen las ROIs que se evalúan en el sistema, en las cuales se encuentran los candidatos a ser clasificados como personas. Para ello se continua con la etapa de extracción de características, en la cual se parte de las ROIs obtenidas y se recibe a su salida un vector cuyas componentes representen las características extraídas de la persona.

3.5 Extracción de características

La correcta extracción de características resulta vital para obtener un sistema fiable. Esta etapa parte de la información de profundidad la cual ha sido procesada para eliminar la mayor cantidad de ruido posible, y a su salida se obtiene un vector, el cual aporta información sobre las características que se desean extraer de la imagen, los métodos para extraer tales características se denominan descriptores. Existen diversos descriptores, desde los mas básicos que extraen información sobre los bordes, forma o color, hasta otros mas sofisticados que analizan las características de forma piramidal, evaluando no solo la imagen de forma elemental, sino tratando la imagen como un conjunto de ventanas interrelacionadas entre sí.

En este Trabajo Final de Grado se han analizado y evaluado dos descriptores, los descriptores de profundidad para reconocimiento de objetos, que se trataron en la sección 2.5.1 y los descriptores de profundidad para la detección de personas en imágenes cenitales, que se trataron en la sección 2.5.2, a continuación se muestra el análisis realizado sobre ellos.

3.5.1 Descriptores de profundidad para reconocimiento de objetos

Los descriptores de profundidad para reconocimiento de objetos se describen en el trabajo [7]. Como ya se ha comentado en el apartado 2.5.1 incluyen diferentes descriptores para imágenes de profundidad y RGB. En el caso de este trabajo se han implementado los descriptores de imágenes de profundidad con el objetivo de determinar su validez para la detección de personas.

En su artículo, [7], se emplean estos descriptores para la posterior clasificación de alimentos, como manzanas o plátanos, en nuestro caso los descriptores se aplican a imágenes de personas observadas

desde una posición cenital. Por tanto una vez obtenido el descriptor se ha de clasificar entre persona y no persona.

El primer paso ha sido la extracción de los descriptores de un conjunto de imágenes de entrenamiento previamente seleccionadas. A partir de dichos descriptores, y el etiquetado manual de las secuencias, se ha entrenado un clasificador SVM. Como ya se ha comentado en la introducción, esta etapa de entrenamiento se realiza offline.

Por ello se comienza leyendo las secuencias de entrenamiento, para cada una de ellas se elimina el ruido con el método comentado en el capítulo 3.3. Tras lo cual se lee de igual forma el archivo *.z16* asociado a la misma secuencia, donde se indica para cada *frame* cuantas personas hay, así como las coordenadas de 6 puntos característicos de cada una de ellas. Con esta información se determina una ROI. Dado que el lenguaje de programación empleado es C++, y que se trabaja con las librerías de visión de OpenCv, las imágenes se representan como matrices, ello implica que cada píxel queda determinado por la fila, columna y valor de la matriz en esta posición. Y finalmente la ROI seleccionada se almacena en una variable matriz para ser procesada y obtener el descriptor correspondiente. .

La matriz es procesada para la obtención del descriptor, para lo cual se hace uso de una serie de ventanas de píxeles, es decir se evalúa la imagen como un subconjunto de ventanas. Cada ventana pasa por un proceso por el cual se le aplica un *kernel* para obtener las características de esa subregión. También se emplean *piramidal efficient match kernels* (EMK) para agregar *kernels* locales en los diferentes niveles de características de la imagen. El uso de esta técnica produce un robustecimiento del sistema. En el capítulo 2.5.1 se explican en profundidad los descriptores empleados.

A la salida del proceso de extracción de características se obtiene el descriptor, un vector compuesto por un conjunto de valores representativos de las características extraídas, denominadas componentes del vector de características. Cada una de las componentes aportan información de forma, bordes, color, morfología de la nube de puntos, etc. En el caso de este descriptor, el vector de características proporcionado consta de 14000 componentes, un número elevado, pero justificado dado el amplio análisis que se realiza de las imágenes de entrada.

En esta fase de entrenamiento el objetivo es obtener vectores de características de las diferentes clases que se evalúan, en este caso se trata de la clase persona y no persona, por lo que obtenemos numerosos vectores de ambas clases. Las imágenes que se evalúan de la base de datos son lo mas diferentes posibles, para que se evalúen el máximo de posibilidades. Una vez obtenida la base de datos de las imágenes, y su posterior conjunto de vectores de características se centra el estudio en la similitud de los vectores de las dos clases.

En un caso óptimo los vectores de características de las dos clases tendrían valores claramente diferentes, para que en su posterior clasificación sea muy discriminante la pertenencia a una clase o a otra y que se cometan el menor número de fallos posibles. Pero dado que nos encontramos ante un sistema real, se ha de contar con la existencia de similitudes. No todos los descriptores valen para cualquier posición de la cámara, tipo de imágenes, entornos, etc, sino que cada uno puede ser empleado en un tipo de sistema diferente.

A continuación se muestran algunos ejemplos de vectores de características obtenidos, así como la ROI de la que se han extraído, 3.5 y 3.6.

Como se aprecia en las imágenes, de forma visual los vectores de persona y no persona en muchos casos son muy similares. Este hecho es negativo, ya que no se podría clasificar de forma correcta. Para evaluar la similitud de los vectores de características de las dos clases estudiadas se procede a estudiar su correlación. Para ello se calcula la varianza 3.4, covarianza 3.5 y el índice de correlación 3.6 entre las dos

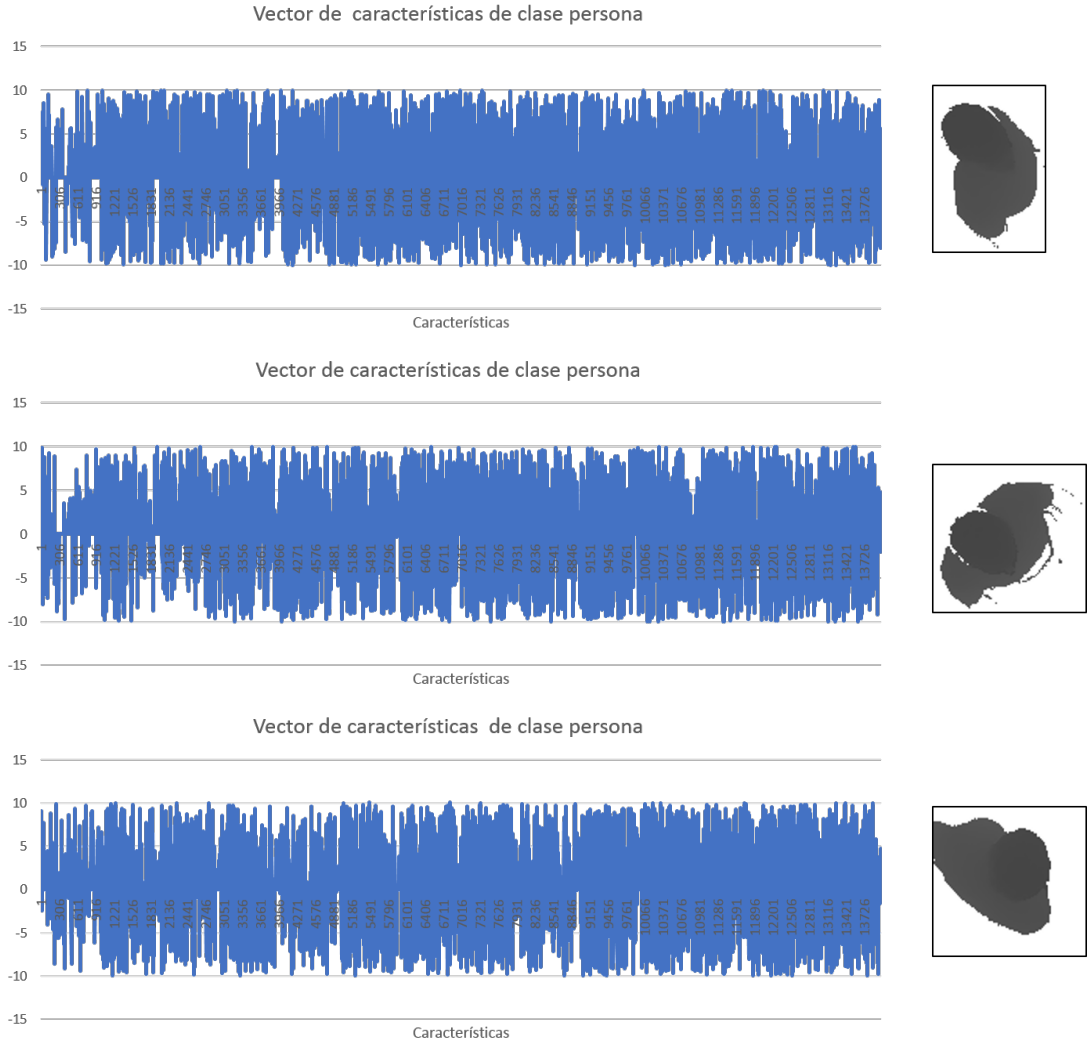

Figura 3.5: Vectores de características de regiones de interés clasificadas como personas.

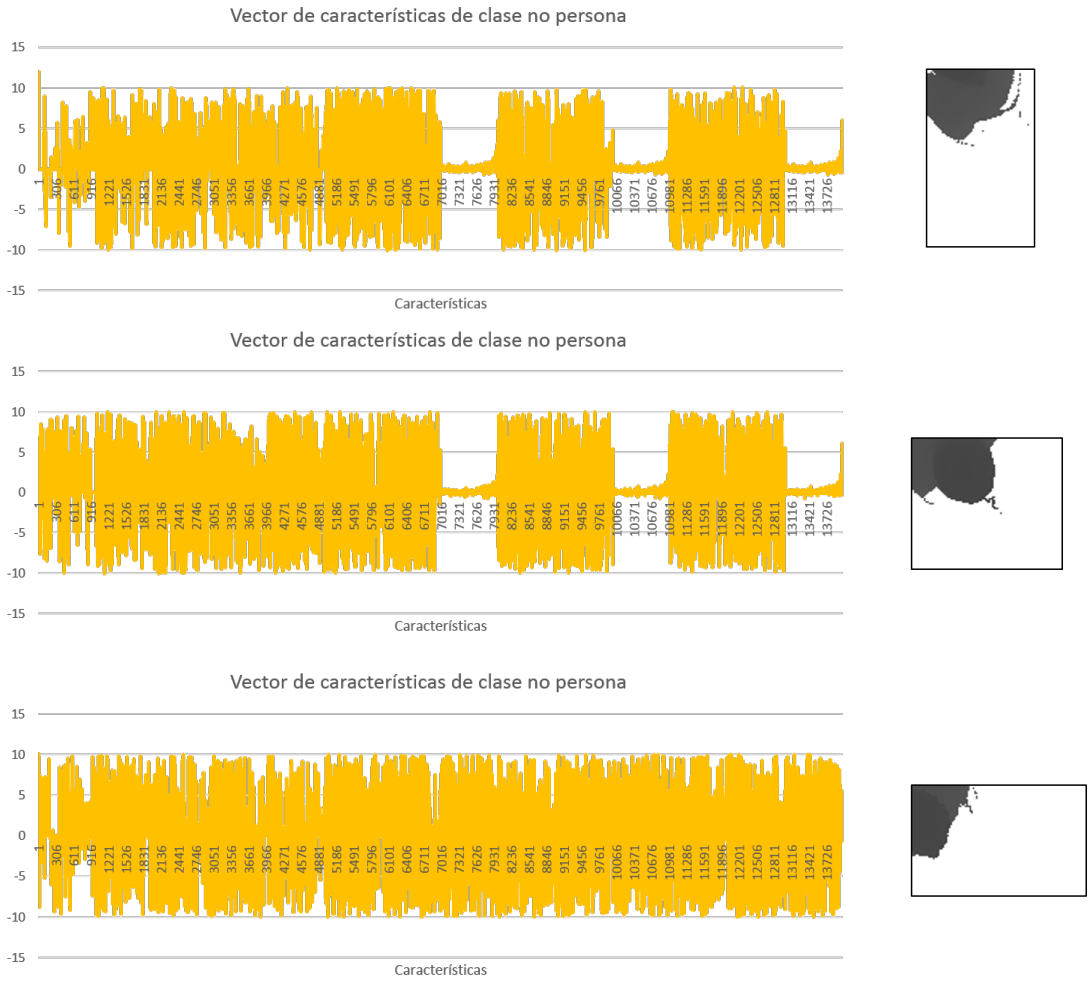

Figura 3.6: Vectores de características de regiones de interés clasificadas como no personas.

clases consideradas para un conjunto de n muestras. Los resultados obtenidos se muestran gráficamente en las figuras 3.7, 3.8, 3.9 y 3.10.

$$\text{Varianza} = \sigma_n^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{3.4}$$

$$\text{Covarianza} = COV(x,y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n} \tag{3.5}$$

$$\text{Indice de correlación} = \rho_{xy} = \frac{COV(xy)}{\sigma_x * \sigma_y} \tag{3.6}$$

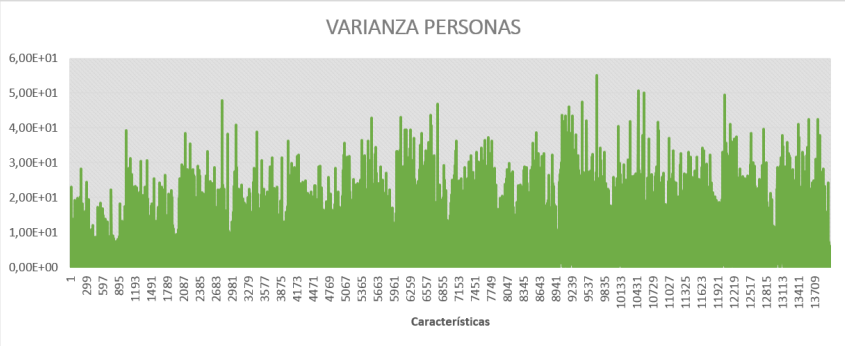

Figura 3.7: Varianza de las 14000 características de los vectores de características de regiones de interés clasificadas como personas.

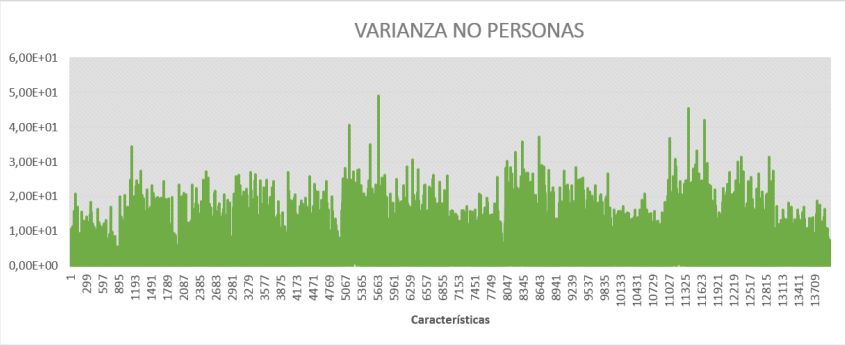

Figura 3.8: Varianza de las 14000 características de los vectores de características de regiones de interés clasificadas como no personas.

Como se puede observar las características que se extraen de las ROIs empleando este descriptor no son lo suficiente discriminantes como para poder realizar una clasificación correcta.

Tras el análisis de la varianza de las componentes de un vector genérico de cada una de las clases, se ha observado que las características extraídas no resultas discriminantes entre la clase persona y no persona. Para obtener un indicador cuantitativo de lo discriminantes o no que resultan los descriptores para este sistema en concreto, se procede a calcular el Indice de Fisher.

El Indice de Fisher fue desarrollado por R.A. Fisher entre 1920 y 1930, es un indicativo de la varianza y la correlación de variables. En la ecuación 3.7 se muestran los pasos para su obtención.

$$\text{Ratio de Fisher} = \frac{(Media(A) - Media(B))^2}{(\sigma_A + \sigma_B)^2} \tag{3.7}$$

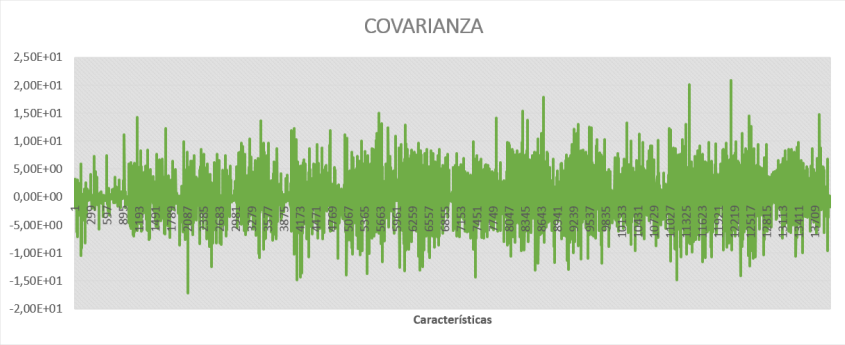

Figura 3.9: Covarianza de las 14000 características de los vectores de características de regiones de interés evaluadas.

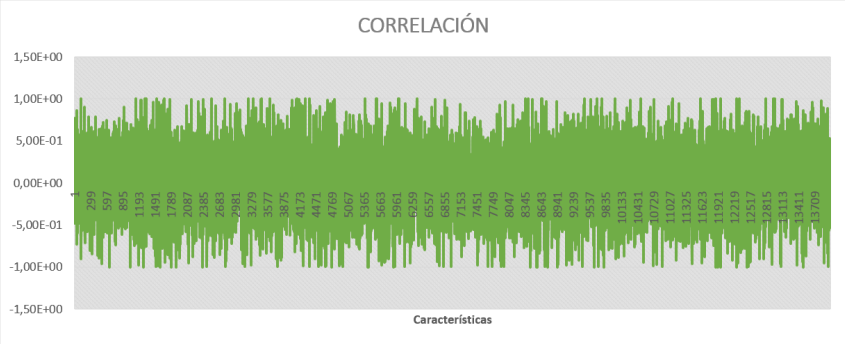

Figura 3.10: Indice de correlación de las 14000 características de los vectores de características de regiones de interés evaluadas.

En la imagen 3.11 se muestran los valores del Ratio de Fisher obtenidos para las 14000 características de los descriptores.

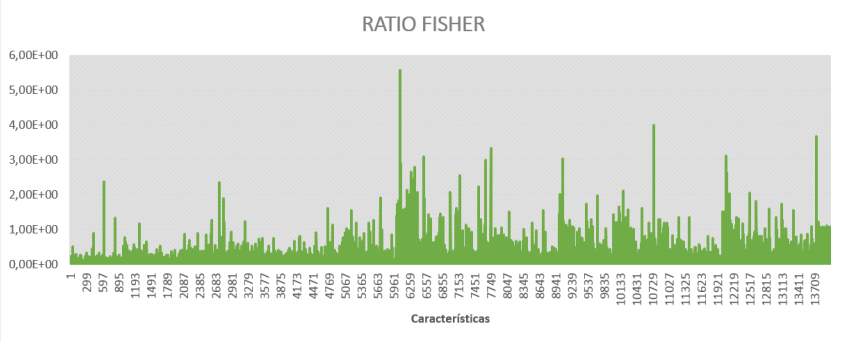

Figura 3.11: Ratio de Fisher de las 14000 características de los descriptores.

Tras el análisis de los vectores de características de la clase persona y no persona se ha obtenido un valor medio para el Indice de Fisher. El valor obtenido es 0,125 y al ser tan bajo indica que las características extraídas por los descriptores no son lo suficientemente discriminantes para lograr una correcta clasificación.

3.5.1.1 Reducción de la dimensionalidad del vector de características

Como se comprobó mediante el Indice de Fisher, las 14000 características que se extrajeron de las imágenes no fueron lo suficientemente discriminantes para poder diferenciar entre clase persona y no persona. Dado

que la dimensión del vector de características es de 14000 componentes, se contempla la posibilidad de que alguna de estas componentes no aporten información relevante para este sistema, ya que entre las 14000 componentes podemos encontrar componentes linealmente independientes y otras dependientes entre sí. Para poder reducir la dimensionalidad del vector, pero manteniendo la mayor parte de la información relevante, se propuso el uso de la técnica PCA, la cual se comentó en la sección 2.6.2. Gracias a la técnica del Analisis de las Componentes Principales se pueden obtener sistemas dimensionalidad reducida manteniendo las componentes que no estén correladas o que lo estén en mayor medida, manteniendo en todo momento la mayor cantidad de información posible.

En el sistema se han evaluado las 14000 componentes, así como el valor de los autovalores asociados a los autovectores de la matriz de correlación. Tras ello se ha comprobado que no existen componentes principales que resulten discriminantes y reduzcan la dimensionalidad del vector de características, por lo que los descriptores implementados no permitan discriminar entre la clase persona y no persona, esto se debe a que todas las componentes contienen información relevante al estar muy correladas, como ya se determinó con el Indice de Fisher.

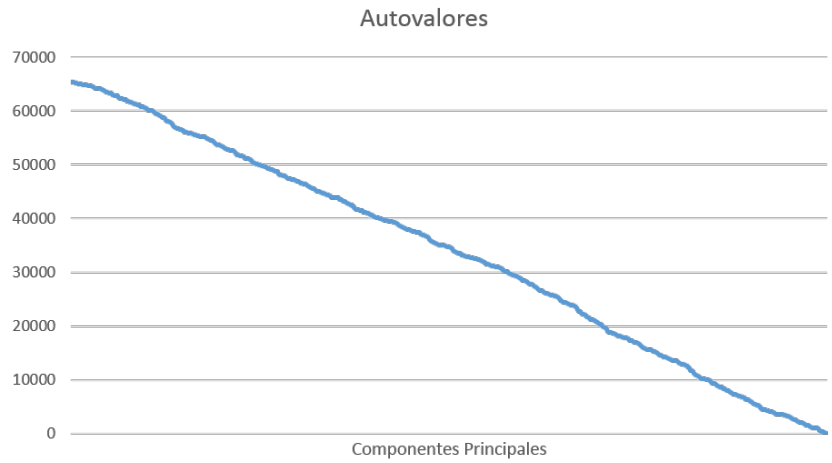

Figura 3.12: Autovalores de las componentes principales evaluadas.

Tras el estudio intensivo de los descriptores desarrollados por [7] implementados para el sistema que se desarrolla en este trabajo, se ha comprobado que no resultan eficientes para este sistema, esto se debe a que no extrae las características correctas para que se pueda determinar si la ROI evaluada pertenece a la clase persona o no persona. Existen diversos factores por los que no resulta empleable en este sistema, en [7] se comenta que ya se realiza una reducción de la dimensionalidad de las componentes del vector de características proporcionado, por lo que dado que sus clases son diferentes a las que se tratan en este TFG puede haber ocultado las características que resultaban tener importancia para este sistema. También cabe destacar que los descriptores que se emplean han sido desarrollados para unas clases concretas, en su caso manzanas, plátanos, botellas, móviles y otros. En todo momento la base de datos que se emplea en su articulo cuenta con imágenes nítidas con bordes muy claros, en nuestro caso contamos con imágenes en posición cenital con un alto contenido de ruido en los bordes de la persona, por tanto tras el análisis de estos descriptores se deduce que no resultan idóneos para el sistema implementado.

3.5.2 Descriptores de profundidad para la detección de personas en imágenes cenitales

Como se ha comentado en el apartado anterior, se han estudiado los descriptores implementados en [7]. Sin embargo, dado que no son válidos para el objetivos de este trabajo, se procede a implementan de los descriptores desarrollados por el grupo de investigación GEINTRA, dentro del departamento de Electrónica de la Universidad de Alcalá.

Estos descriptores tienen como entrada las ROIs seleccionadas a partir del conjunto de máximos detectados, cuyo calculo se trata en el capitulo 3.4.

Como se trata en profundidad en la sección 2.5.2, estos descriptores han sido desarrollado específicamente para la extracción de características de personas en imágenes de profundidad, tomadas desde una cámara de tiempo de vuelo en situada en posición cenital. Los vectores de características obtenidos se basan en la densidad de puntos que encontramos en unos niveles que se definirán más adelante, por lo que la magnitud de cada característica variará con fisonomía de la persona.

El procedimiento seguido para la obtención de los descriptores se trató en 2.5.2, en la imagen 3.13 se muestran los vectores de 20 componentes de dos personas, mientras que en la imagen 3.14 se muestran ejemplos del vector de 6 componentes más robusto, el cual es la entrada del clasificador.

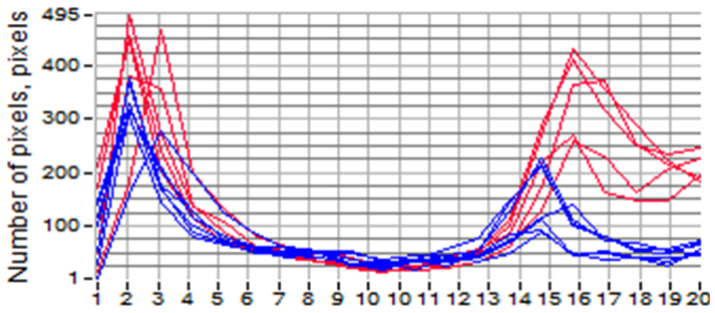

Figura 3.13: Ejemplos de diferentes vectores de características, los vectores de color rojo representan a una persona de 180cm y pelo corto, mientras que los de color azul representan a una persona de 162cm y pelo largo.

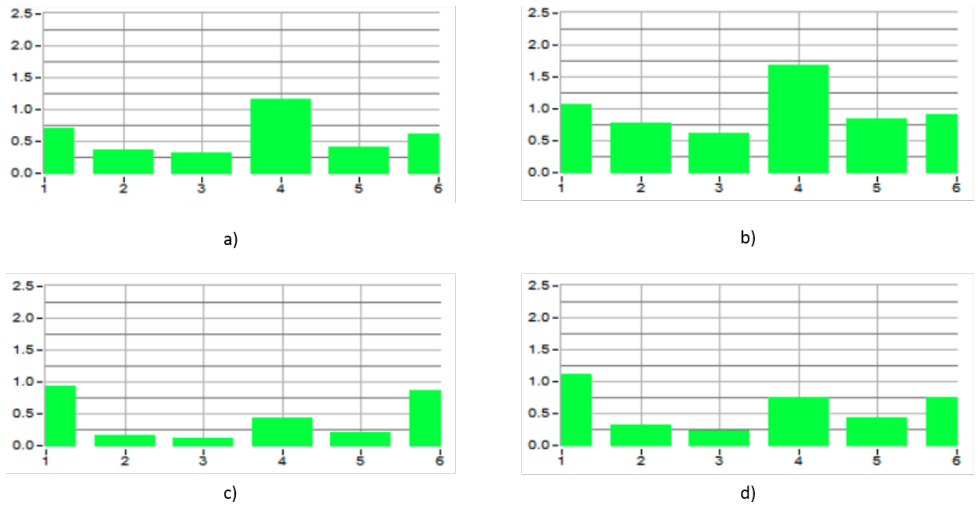

Figura 3.14: Ejemplos del vector de 6 componentes obtenido a partir de el vector previo de 20 componentes, cuya dimension se reduce para aumentar la robustez. El vector a) corresponde a una persona de 165cm, b) a una de 183cm, c) a una de 187cm y d) a una de 197cm de altura.

3.6 Sistema de clasificación

Una vez realizado el proceso de extracción de características, la siguiente etapa desarrollada es la clasificación. Como se comentó anteriormente, el sistema cuenta con 2 etapas, una online y otra offline, en la etapa offline se realiza el entrenamiento del clasificador utilizado. Para ello se parte de una base de datos de imágenes de profundidad grabadas en el espacio inteligente (ISPACE) del grupo GEINTRA, etiquetada de forma manual. Esta información permite conocer la pertenencia a las clases de cada imagen evaluada. Los descriptores extraídos para las imágenes de entrenamiento, junto con la información del etiquetado permiten entrenar un clasificador. En este caso se trata de un clasificador basado en la técnica PCA, la cual se explica en la sección 2.6.2.

Se parte los vectores obtenidos de la base de datos, en este caso se emplearon 2955 *frames* para el entrenamiento, en los cuales se encontraban 12 personas con características diferentes, ver tabla 3.1, tras ello se separan en función de la clase a la que pertenezcan, persona o persona con complemento (sombrero, gorra, etc.). A fin de elaborar la matriz de transformación de las dos clases mencionadas anteriormente, se han empleado secuencias con una única persona. Para cada una de las clases se realiza la búsqueda de las componentes principales más representativas, es decir las que aporten mayor información, manteniendo la incorrelación entre ellas. Una vez obtenidas tales componentes, en este caso se emplearon las 10 primeras componentes principales con autovalores de mayor valor, se procede a formar la matriz de transformación de cada una de las clases a partir de los autovalores y autovectores seleccionados para cada clase. En el caso de la clase de persona se emplearon 2088 vectores de características para la determinación de la matriz de transformación, y en el caso de la clase de persona con accesorio 748 vectores de características. Con ello ya se tiene el clasificador entrenado, y se puede proceder a la clasificación en la etapa online.

Personas presentes en las secuencias de entrenamiento PCA
Hombre de complexión media, pelo corto y altura media
Hombre de complexión grande, pelo largo y altura media
Hombre de complexión grande, pelo corto y altura alta
Hombre de complexión pequeña, pelo corto y altura baja
Hombre de complexión media, pelo corto y altura media
Mujer de complexión pequeña, pelo largo y altura alta
Mujer de complexión pequeña, pelo largo y altura baja
Mujer de complexión pequeña, con un sombrero grande y altura media
Mujer de complexión pequeña, con un sombrero pequeño y altura media
Hombre de complexión media, con gorra y altura media
Hombre de complexión media, con un sombrero grande y altura media
Hombre de complexión media, con un sombrero pequeño y altura media
Puños
Sillas en movimiento

Tabla 3.1: Personas empleadas en el entrenamiento del clasificador PCA.

Una vez realizado el entrenamiento, en la etapa online se realiza el procesado de las imágenes de entrada para detección y conteo de personas. Para ello, por cada imagen de entrada, tras filtrar las imágenes con el objetivo de reducir el ruido, se obtienen los máximos locales, se determina la ROI a analizar alrededor de cada máximo. Para cada una de las ROIs se extrae un descriptor de 6 características que posteriormente se clasifica como persona o no persona utilizando PCA. Para ello se proyecta cada vector de características extraídos a través de las dos matrices de transformación de la etapa offline, ver ecuación 3.8. Una vez transformado el vector a través de las dos matrices de transformación se procede a su recuperación, ver ecuación 3.9, para obtener el vector de características en el espacio inicial, pero su valor ya no es el inicial, sino que se produce una diferencia entre sus valores, tal diferencia denominada error de recuperación,

ε_{PCA}, indica el grado de pertenencia a cada una de las clases. Siendo más evidente la pertenencia a una clase cuanto menor es tal error de recuperación.

$$\hat{\Phi} = U\Omega \tag{3.8}$$

$$\varepsilon_{PCA} = \|\Phi - \hat{\Phi}\| \tag{3.9}$$

Una vez finalizada la clasificación el sistema de detección de personas básico estaría completo, es decir una vez entrenado el sistema con la etapa offline, se puede proceder a trabajar con el sistema en tiempo real. El procesamiento en la fase online realiza una adquisición de imágenes para cada *frame*, selecciona la ROI que se ha de evaluar, elimina el ruido, extrae el vector de características de cada ROI que se ha de evaluar y clasifica la ROI como persona o no persona. Por ello el detector y contador de personas está realizado, pero el sistema realizado puede mejorarse, ya que una función básica de estos sistemas consiste en conocer si las personas detectadas permanecen en el entorno o se trata de personas nuevas, también es de gran interés la trayectoria de la persona, tanto para el conteo de personas como para la detección de aglomeraciones en ciertos entornos. Por ello en el siguiente capítulo se explica el sistema de seguimiento (tracking) implementado imprescindible para el conteo de personas.

3.7 Sistema de seguimiento

Como se comentó anteriormente a la entrada de la etapa de clasificación en tiempo real contamos con todos los posibles candidatos a ser clasificados como persona, y tras ella ya contamos con los que pertenecen a tal clase, estos positivos serán los que se tengan en cuenta para el sistema de seguimiento. El sistema de seguimiento que se desea implementar ha de cumplir ciertas restricciones, como detectar si una persona ya estaba presente en *frames* anteriores, evaluar su trayectoria o su velocidad.

Existen multitud de sistemas de seguimiento, como el Filtro de Kalman, pero para este sistema se ha seleccionado el Filtro de Partículas Extendido debido a su carácter multimodal. La principal ventaja de este filtro consiste en que con un solo seguidor es capaz de evaluar multitud de individuos, mientras que otros sistemas como el Filtro de Kalman requieren implementar uno para cada individuo evaluado.

En la sección 2.7.2 se explica en profundidad el funcionamiento y las etapas de un filtro de partículas extendido convencional, pero en este sistema se realizan ciertas modificaciones.

3.7.1 Etapas del Filtro de Partículas Extendido implementado

El sistema de seguimiento implementado recibe como entrada las imágenes adquiridas y la posición dentro de la imagen de las posibles personas detectadas tras el proceso de clasificación. A continuación se muestran las etapas por las que pasa el sistema de seguimiento.

- Adquisición de la imagen del *frame* actual (en tiempo real) y lectura de la salida del sistema de clasificación, el archivo *.result*, el cual contiene el número de personas detectadas y la localización de su máximo dentro del entorno.
- Reducción del ruido de la imagen siguiendo el procedimiento del apartado 3.3.
- Una vez obtenida la imagen con el mayor ruido posible eliminado se obtienen $N_{Partic_{INICIAL}}$ partículas de cada persona que el clasificador determinó que se encontraba en el entorno, pero ha de cumplir la condición de que el error de recuperación de esta persona se encuentre dentro de los

margenes normales, entre 0 y 0,5 para personas sin sombrero y entre 0 y 1 para las personas con sombrero.

- Posteriormente comienza el proceso del filtro de partículas en profundidad, al cual se introducirán como datos de entrada principales las personas detectadas y $N_{Particic_{INICIAL}}$ partículas asociadas a cada una de ellas.

- Dentro del código realizado del XPFCP se emplean diversas estructuras cuyo contenido es de suma importancia para entender el proceso, una de ellas es el *XpfCluster*, dentro de la cual se encuentran:*identify*, *candidate*, *validated*, *nMembers*, *centroid* y *members*.

A fin de describir el funcionamiento de filtro de partículas implementado, se presenta a continuación las características básicas del filtro en una iteración cualquiera.

En primer lugar al filtro recibe información de la etapa de clasificación, de la cual obtiene el número de personas, su centroide y un número determinado de partículas de cada una de ellas. Tras ello se realiza una asociación de tales personas con los *clusters* presentes en el filtro, esta asociación esta basada en la distancia euclidea entre los centroides de ambos. El hecho de recibir la información de las personas a partir de la etapa de clasificación del sistema, evita que se contemple la etapa de clasificación inicial de un filtro de partículas extendido con proceso de clasificación.

Una vez contempladas las partículas de todas las personas presentes en la escena se realiza un sesgo, añadiendo al filtro un porcentaje determinado del total de partículas del mismo. En la ecuación 3.10 se muestran el porcentaje de partículas nuevas que se introducen en cada etapa, $N(\,\%)_{PARTIC_{NEW}}$, así como el de las partículas que se almacenan para la siguiente iteración, $N(\,\%)_{PARTIC_{SAVE}}$, donde $N_{PARTIC_{NEW}}$ es el número de partículas añadidas y $N_{PARTIC_{SAVE}}$ es el número de partículas almacenadas para la iteración posterior. Tanto el parámetro $N(\,\%)_{PARTIC_{NEW}}$ como el número de partículas totales, N_{PARTIC} se puede variar para la ejecución del sistema.

$$
\begin{aligned}
N(\,\%)_{PARTIC_{NEW}} &= \frac{N_{PARTIC_{NEW}}}{N_{PARTIC_{NEW}} + N_{PARTIC_{SAVE}}} = \frac{N_{PARTIC_{NEW}}}{N_{PARTIC}} \\
N(\,\%)_{PARTIC_{SAVE}} &= \frac{N_{PARTIC_{SAVE}}}{N_{PARTIC_{NEW}} + N_{PARTIC_{SAVE}}} = \frac{N_{PARTIC_{SAVE}}}{N_{PARTIC}} \\
N(\,\%)_{PARTIC_{NEW}} &+ N(\,\%)_{PARTIC_{SAVE}} = 100\,\% \\
N_{PARTIC} &= N_{PARTIC_{NEW}} + N_{PARTIC_{SAVE}}
\end{aligned}
\tag{3.10}
$$

Tras la inclusión de las $N_{PARTIC_{NEW}}$ en el filtro, se cuenta con el total de las partículas, por lo que se comienza la etapa de asignación del peso a cada una de las partículas. En el filtro implementado, el peso que se asocia a cada partícula esta directamente relacionado con la distancia al centroide del *cluster* al cual pertenece la partícula. Una vez que todas las partículas cuentan con un peso asociado, se seleccionan las $N_{PARTIC_{SAVE}}$ que se almacenan para la iteración posterior en función de tal peso.

A continuación se realiza la etapa de clasificación final, para la cual se emplea un algoritmo k-medias para la determinación del número de personas y su centroide, partiendo del conjunto de partículas $N_{PARTIC_{SAVE}}$. Las características mas importantes del algoritmo empleado se exponen a continuación.

- El algoritmo k-medias empleado cumple los requisitos que estableció Mac Queen en 1967, donde las

características empleadas han de ser discriminantes y estar incorreladas, un solo datos solo puede estar asociado a un cluster y los cluster vacíos se eliminan para las siguientes iteraciones.

- El algoritmo k-medias parte de un conjunto (Y) de $N_{PARTIC_{SAVE}}$ partículas que se agruparan en N-1 clases.

- Las clases en las que se agrupan se denominaran cluster, y cada cluster representara una persona, teniendo por tanto el centroide de las particulas que pertenencen al cluster y por tanto de la persona clasificada.

- En el diagrama 3.15 se muestran la diferentes etapas que conforman el algoritmo k-meas extendido que se ha empleado.

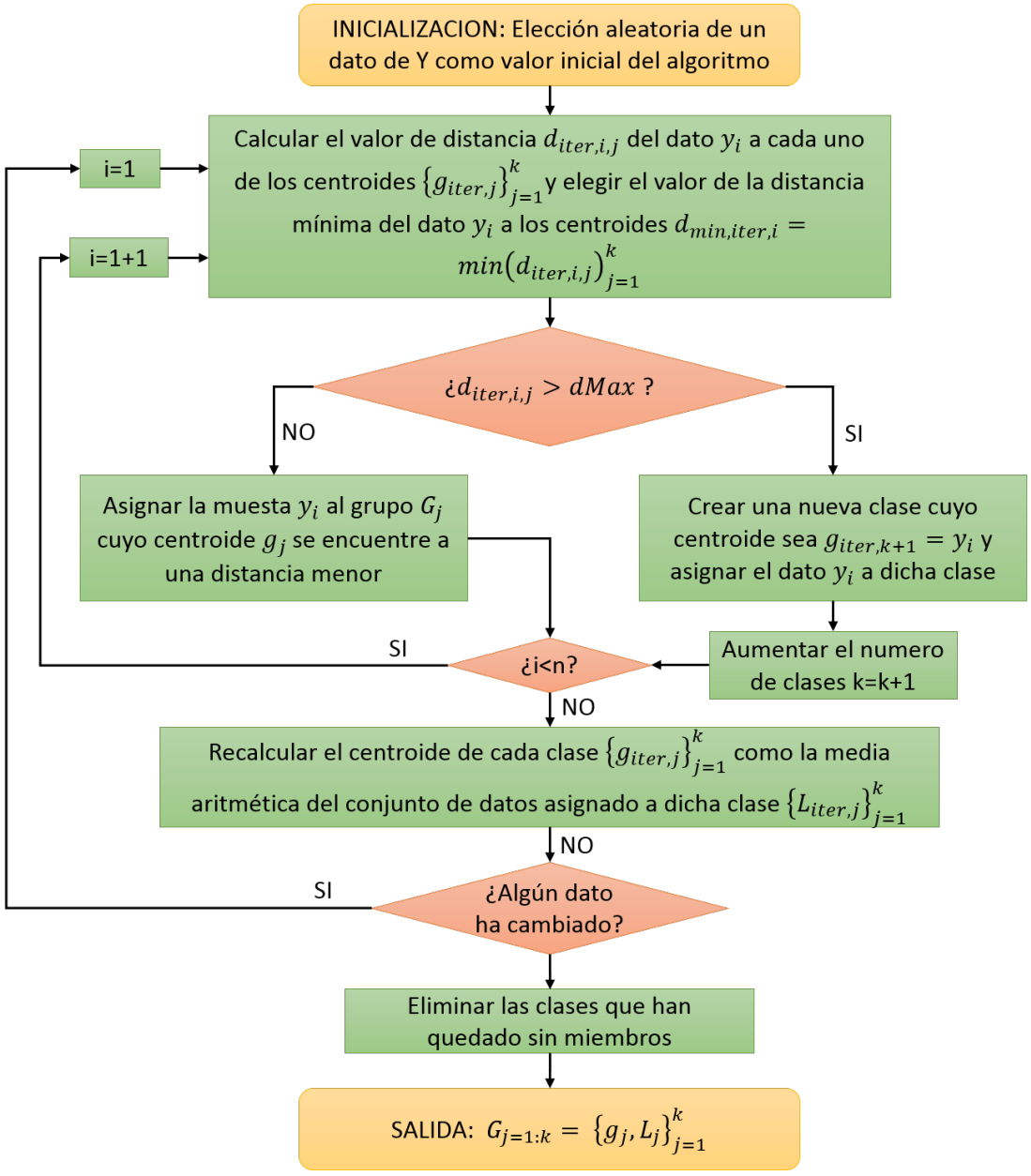

Figura 3.15: Diagrama de bloques del algoritmo k-medias extendido empleado en la etapa de clasificación de las partículas del sistema.

Tras esta última etapa se obtiene a la salida las personas clasificadas, tras lo cual se procede a realizar el histórico presente las características de las personas presentes en el entorno. En las asociaciones por distancia empleadas en todo el filtro se parte de una distancia maxima contemplada, la cual se toma como parámetro del filtro de partículas, que junto con $N(\%)_{PARTIC_{NEW}}$ y N_{PARTIC}, sera evaluado su efecto sobre los resultados obtenidos debido a su influencia en el filtro desarrollado.

El sistema de clasificación y seguimiento se ha completado en gran medida, pero para poder aportar más información al proceso se ha realizado un histórico de los datos del sistema, a continuación se relatan los procesos que se han llevado a cabo para su consecución.

El proceso comenzó con dos archivos, uno de extensión *.z16* que contiene la información en si procesada por la cámara, y otro de extensión *.result* que nos indica las personas detectadas por el sistema y que han sido clasificadas como persona. A la salida del sistema de seguimiento se pueden encontrar cluster que representen a persona que estaban al comienzo del filtro, personas que no lo estaban, inclusive personas que hayan desaparecido. Se ha realizado un proceso de comparación entre la entrada y salida del sistema de seguimiento, lo que aporta un histórico, para la consecución del histórico se han llevado dos procesos, uno de asociación entre entrada y salida, y finalmente el histórico en sí. A continuación se muestran los algoritmos de ambos procesos, 3.1 y 3.2.

Data: Fichero de extensión *.result* que contiene el centroide de los candidatos previamente clasificadas en la clase persona, y también se cuenta con los cluster de salida del sistema de seguimiento

Result: Asociación cada una de las I personas que se encontraban en cada C *frame* de la entrada, A, con las J del mismo C *frame* a la salida del sistema,B.

for C *valido* $\in A$ **do**
- **for** $i \in I$ **do**
  - $Distancia_{ij}$ = distancia mínima entre i y las J posibles personas en el *frame* C en B);
  - **if** $Distancia_{ij} < Distancia_{MAX}$ **then**
    - Se asocia la persona $i \in I$ con la $j \in J$ adoptando el ID (variable *identify* de los cluster) de j;
    - La persona $j \in J$ no se contempla en los siguientes *frames*, C;
    - Índice de asociación = 0;
  - **else**
    - La persona no se encuentra asociada a la salida con ningún ID;
    - Índice de asociación = 2;
- **for** *($j \in J$) and (j permanezca sin asociar)* **do**
  - La persona no se encuentra asociada a la salida con ninguna entrada pero mantiene su ID;
  - Índice de asociación = 0.5;

Algoritmo 3.1: Algoritmo del código empleado para la asociación de las personas entre la entrada y salida del sistema de seguimiento

Tras la ejecución del algoritmo 3.1, ya se cuenta con las personas asociadas, y en su defecto se cuenta con las personas que solo se encuentran a la salida del sistema de seguimiento o a su entrada. El único caso en el cual la persona no esta identificada en un *frame*, es cuando están a la entrada pero sin asociarse con ninguna a la salida. Esto se debe a que el Filtro de Partículas extendido que se emplea, trabaja con estructuras del tipo *XpfCluster*, las cuales contienen un identificador, *identify*, mediante el cual asocian partículas a los cluster de varios *frames* y se mantiene una asociación interna. Como se comentó en la sección 3.7.1 también se cuanta con las variables *candidate* y *validated*, que restringen la validación de una persona en la imagen hasta que no se encuentran presentes varios *frames* consecutivos en el entorno.

Pero tras ello se ha de corregir los *frames* que se desestimaron, ya que era correcta la hipótesis de que la persona se encontrase en el entorno. Para ello se implementa el algoritmo 3.2, donde contamos con las personas con índice de asociación, $ID_{Asociacion}$, con valor 0, 0.5 y 2, que indican su procedencia.

Data: Para cada *frame* $f \in F$ (C=*frames* con personas) se cuenta con las personas, P, que se encontraban a la entrada y/o salida del sistema de seguimiento, $ID_{Asociacion} = 0, 1o2,$, con sus respectivos centroides, y en su caso con su ID de procedencia del Filtro de partículas extendido.

Result: Fichero con extensión *.xpf* con el histórico de las personas en el sistema.

for $f \in F$ **do**
- **for** $p \in P$ *con* $ID_{Asociacion} = 2$ **do**
  - **for** $inc < 3$ **do**
    - $Distancia_{MIN}$ = distancia mínima entre p y las posibles personas con $ID_{Asociacion} \neq 2$ pertenecientes al *frame* $c - f$;
    - **if** $Distancia_{MIN} < Distancia_{MAX}$ **then**
      - Se asocia la persona p con la persona perteneciente al *frame* $c - f$ que cumple la condición de distancia, adoptando el ID de tal persona (variable *identify* de los cluster);
      - A su vez el $ID_{Asociacion}$ de la persona p se modifica a 1;
    - **else**
  - **for** $inc < 3$ **do**
    - $Distancia_{MIN}$ = distancia mínima entre p y las posibles personas con $ID_{Asociacion} \neq 2$ pertenecientes al *frame* $c + f$;
    - **if** $Distancia_{MIN} < Distancia_{MAX}$ **then**
      - Se asocia la persona p con la persona perteneciente al *frame* $c - f$ que cumple la condición de distancia, adoptando el ID de tal persona (variable *identify* de los cluster);
      - A su vez el $ID_{Asociacion}$ de la persona p se modifica a 1;
    - **else**
- **for** $p \in P$ **do**
  - Impresión en un fichero de escritura con extensión *.xpf* el ID (variable *identify* de los cluster), x,y y $altura(cm)$ del centroide e $ID_{Asociacion}$ de cada p ;

Algoritmo 3.2: Algoritmo del código empleado para la consecución del histórico del sistema de seguimiento

Tras la ejecución del algoritmo 3.2 ya se cuenta con el fichero de extensión *.xpf*, el cual permitirá obtener resultados en posteriores iteraciones.

Capítulo 4

Resultados

4.1 Introducción

En este capítulo se presentan los resultados obtenidos por el sistema implementado, como se muestra a continuación se realizan diversos experimentos para la obtención de los parámetros óptimos para el sistema.

4.2 Entorno experimental y base de datos utilizada

Este trabajo se ha realizado en un entorno interior, en el cual la cámara se sitúa a una altura de 3.4m. En las tramas empleadas para el desarrollo del sistema se cuenta con diversas personas, y su circulación por el entorno es variante en todo momento, es decir, pueden encontrarse agrupadas formando aglomeraciones, así como circular con separación. Para ello se parte de una base de datos de imágenes de profundidad grabadas en el espacio inteligente (ISPACE) del grupo GEINTRA, etiquetada de forma manual.

4.3 Resultados obtenidos en función de los parámetros

En el presente trabajo se ha obtenido un sistema de detección y conteo de personas, pero los resultados obtenidos dependen de diversos parámetros, a continuación se realiza un estudio de los resultados obtenidos en función de tales variables.

4.3.1 Métricas de calidad

En primer lugar se describen las medidas que permitirán evaluar la efectividad y fiabilidad del sistema. A la salida del sistema se cuenta con las personas detectadas, de las cuales se conoce si identificador y trayectoria. Dado que la base de datos empleada esta etiquetada de forma manual, se puede conocer la veracidad de los resultados obtenidos.

De la base de datos, se conoce la posición de las personas etiquetadas, y dado que a la salida del sistema también se cuenta con el centroide de las personas detectadas se puede asociar las medidas obtenidas. A continuación se presentan las diferentes posibilidades que se pueden dar:

- **Verdaderos positivos (TP, True Positives):** detecciones del sistema clasificadas como personas y que en la base de datos se asocia con una persona etiquetada manualmente.

- **Verdaderos negativos (TN, True Negatives):** detecciones del sistema clasificadas como no personas y que no se asocian con ninguna persona de la base de datos etiquetada.
- **Falsos positivos (FP, False Positives):** detecciones del sistema como personas que en la base de datos no se asocian con una persona etiquetada manualmente.
- **Falsos negativos (TN, False Negatives):** detecciones del sistema como no persona que en la base de datos se asocian a una persona etiquetada manualmente.

A continuación se detallan los estudios realizados sobre el efecto de los parámetros sobre el resultado obtenido en el sistema, para ello se ha evaluado una secuencia de dos personas como la que se muestra en la imagen 4.1

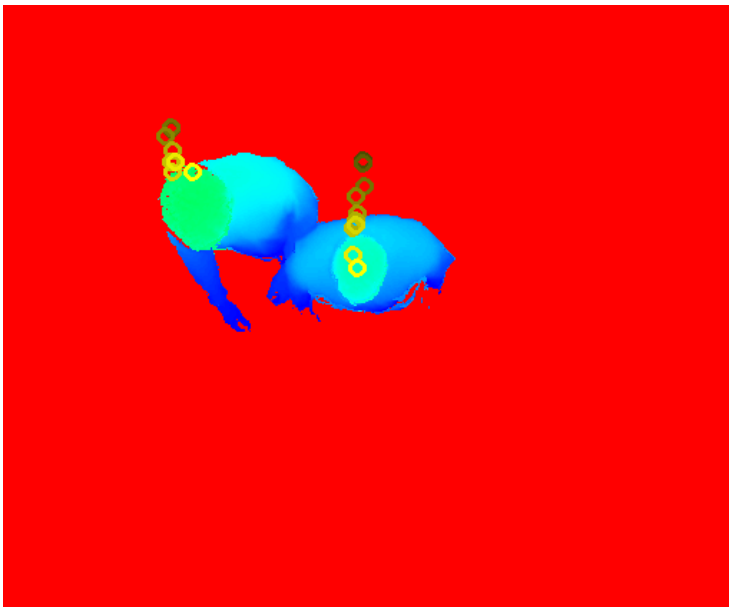

Figura 4.1: Imagen de uno de los frames de la secuencia de empleada para el análisis paramétrico.

4.3.2 Resultados en función del porcentaje de partículas mantenidas entre iteraciones consecutivas

En esta sección se realiza un análisis paramétrico de los resultados obtenidos por el filtro, variando el porcentaje de partículas mantenidas en el filtro de partículas extendido implementado. Como se comentó en la sección 3.6, el filtro de partículas cuenta con un número total de partículas, N_{PARTIC}, de las cuales un cierto número de partículas se añaden en la iteración actual, $N_{PARTIC_{NEW}}$, y otras se conservan de iteraciones anteriores, $N_{PARTIC_{SAVE}}$. En todo momento se cumple la condición de que la suma de $N_{PARTIC_{SAVE}}$ y $N_{PARTIC_{NEW}}$ suman N_{PARTIC}, de tal forma que se establece un porcentaje entre las partículas añadidas y conservadas.

$$N(\%)_{PARTIC_{NEW}} = \frac{N_{PARTIC_{NEW}}}{N_{PARTIC_{NEW}} + N_{PARTIC_{SAVE}}} = \frac{N_{PARTIC_{NEW}}}{N_{PARTIC}}$$

$$N(\%)_{PARTIC_{SAVE}} = \frac{N_{PARTIC_{SAVE}}}{N_{PARTIC_{NEW}} + N_{PARTIC_{SAVE}}} = \frac{N_{PARTIC_{SAVE}}}{N_{PARTIC}} \tag{4.1}$$

$$N(\%)_{PARTIC_{NEW}} + N(\%)_{PARTIC_{SAVE}} = 100\,\%$$

En este caso se a realizado un análisis de los verdaderos positivos (TP), y los falsos negativos (FN) obtenidos en función del valor del parámetro $N(\%)_{PARTIC_{NEW}}$. De tal manera que para cada valor de $N(\%)_{PARTIC_{NEW}}$ evaluado entre 10 % y 90 % se observa el porcentaje de TP entre el total del detecciones, TPR (True Positives Rate), y el porcentaje de FN entre el total de detecciones, FN (False Negatives Rate). En la figura 4.2 se presenta el análisis comentado.

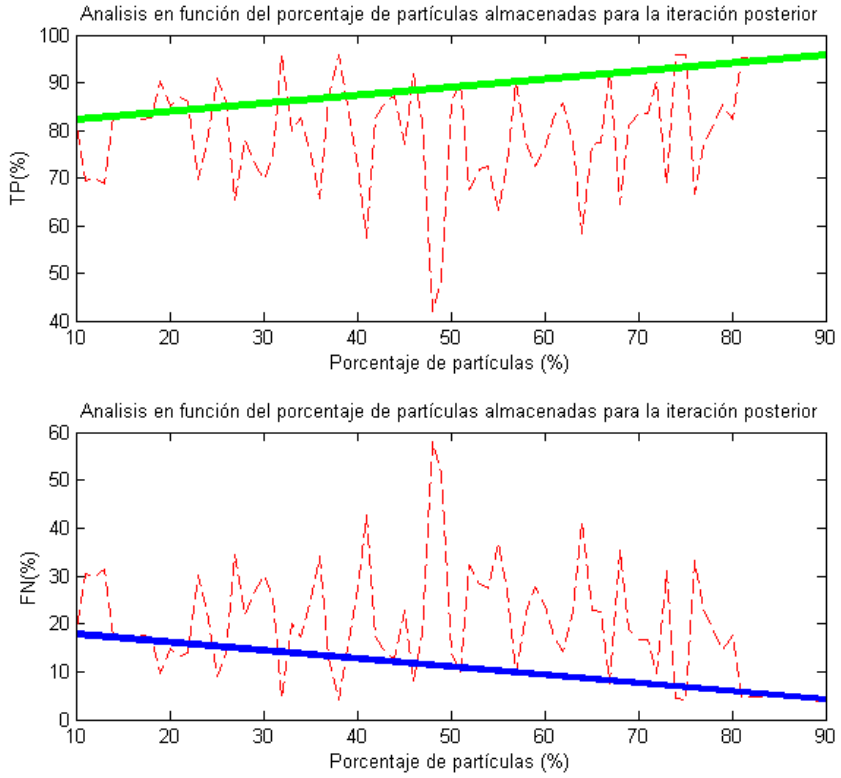

Figura 4.2: Análisis paramétrico en función de $N(\%)_{PARTIC_{NEW}}$ evaluando TP(%) y FN(%).

En la figura 4.2 se muestran en la parte superior en rojo los valores del TPR obtenidos, y en verde los valores de TPR obtenidos tras una interpolación cuadrática con 3 coeficientes. En la gráfica inferior se observa en azul los valores de FNR obtenidos tras una interpolación cuadrática con 3 coeficientes. El hecho de que los datos del TPR y FNR obtenidos (rojo) cuenten con una gran cantidad de ruido se podría reducir, dado que en el caso de realizar el estudio con una mayor cantidad de secuencias se obtendría un resultado mas estable.

De la figura 4.2 se extraen diversas conclusiones, en primer lugar se observa que el porcentaje de partículas que se introducen en cada iteración, $N(\%)_{PARTIC_{NEW}}$, no tiene un gran impacto sobre el sistema. Ello se debe el porcentaje de partículas introducidas en cada nuevo frame, tiene mas influencia sobre el histórico de las personas presentes en la escena, que sobre el hecho de que la persona detectada se encuentre o no presente realmente en el entorno. Se ha observado de forma experimental que entorno a $N(\%)_{PARTIC_{NEW}} = 67\,\%$ se obtienen los mejores resultados en el sistema implementado.

4.3.3 Resultados en función de la distancia de asociación entre clusters

En esta sección se realiza un análisis paramétrico de los resultados obtenidos por el filtro, variando la distancia de asociación entre los cluster del filtro implementado en el sistema. Como se comentó en la sección 3.6, en el filtro se realizan una serie de asociaciones entre las detecciones presentes en el sistema

en la iteración anterior en el instante $t-1$ y los que analizan en el instante t. La asociación se realiza estableciendo una distancia , $Distancia_{MAX}$, en función de la cual se pueden asociar los cluster del instante t con los del instante $t-1$ siempre y cuando no se supere la distancia máxima de asociación.

Al igual que en la sección anterior se a realizado un análisis de los verdaderos positivos (TP), y los falsos negativos (FN) obtenidos, en este caso en función del valor del parámetro $Distancia_{MAX}$. De tal manera que para cada valor de $Distancia_{MAX}$ evaluado entre 50 y 600 se observa el porcentaje de TP entre el total del detecciones, y el porcentaje de FN entre el total de detecciones, los valores de $Distancia_{MAX}$ se dan en unidades pixélicas al cuadrado, ya que la distancia entre los cluster se evalúa como la distancia euclídea al cuadrado, procesando las coordenadas u y v, sin considerar la altura, ver ecuación 4.2. En la figura 4.3 se presenta el análisis comentado.

$$Distancia_{AB} = \sqrt{(v_B - v_A)^2 + (u_B - u_A)^2} \tag{4.2}$$

Donde A y B son los indices de los cluster evaluados, y v_A, u_A, v_B y u_B sus respectivas coordenadas pixélicas.

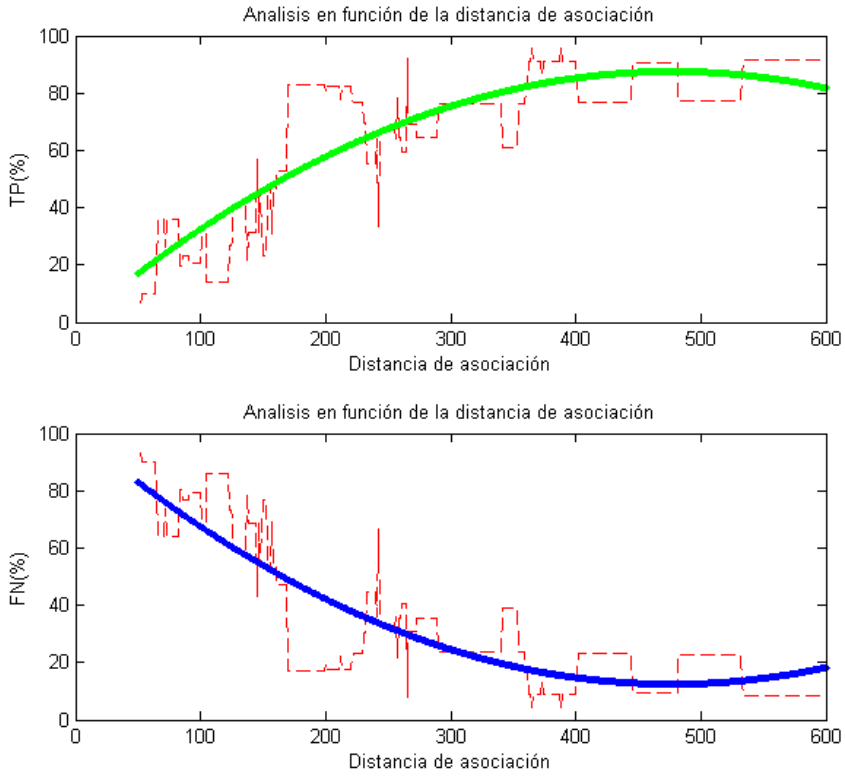

Figura 4.3: Análisis paramétrico en función de $Distancia_{MAX}$ evaluando TP(%) y FN(%).

En la figura 4.3 se muestran en la parte superior en rojo los valores del TPR obtenidos, y en verde los valores de TPR obtenidos tras una interpolación cuadrática con 3 coeficientes. En la gráfica inferior se observa en azul los valores de FNR obtenidos tras una interpolación cuadrática con 3 coeficientes. El hecho de que los datos del TPR y FNR obtenidos (rojo) cuenten con una gran cantidad de ruido se podría reducir, dado que en el caso de realizar el estudio con una mayor cantidad de secuencias se obtendría un resultado mas estable.

De la figura 4.3 se observa que cuanto mayor es la distancia de asociación mayor es el número de

verdaderos positivos detectados, y por ello menor el número de falsos negativos detectados. Este hecho se debe a que cuanto mayor sea esta distancia, mayor es la posibilidad de que dos cluster se asocien y por ello la de que una detección se encentre en la base de datos etiquetada. El hecho de que se asocien mas cluster no es favorable en todos los casos, ya que superado cierto valor de la distancia máxima de asociación se pueden asociar cluster que corresponden a personas diferentes en la base de datos etiquetada. Como se observa en la figura 4.3 a partir del valor 300 para $Distancia_{MAX}$ el porcentaje de TP se normaliza, y dado que el interés de este sistema radica en gran medida en el conteo e identificación de las personas presentes en el entorno, se ha determinado el valor de $Distancia_{MAX} = 300$ como óptimo.

4.3.4 Resultados en función del número de partículas del filtro de partículas extendido

En esta sección se realiza un análisis paramétrico de los resultados obtenidos por el filtro, variando el número de partículas totales del filtro de partículas extendido implementado, N_{PARTIC}. Como se comentó en la sección 4.3.2, el filtro de partículas cuenta con un numero total de partículas, N_{PARTIC}, de las cuales un cierto numero de partículas se añaden en la iteración actual, $N_{PARTIC_{NEW}}$, y otras se conservan de iteraciones anteriores, $N_{PARTIC_{SAVE}}$. Cumpliéndose en todo momento la ecuación 4.1,

En este caso se a realizado un análisis de los verdaderos positivos (TP), y los falsos negativos (FN) obtenidos en función del valor del parámetro N_{PARTIC}. De tal manera que para cada valor de N_{PARTIC} evaluado entre 10 y 12350 partículas se observa el porcentaje de TP entre el total del detecciones, TPR (True Positives Rate), y el porcentaje de FN entre el total de detecciones, FN (False Negatives Rate). En la figura 4.4 se presenta el análisis comentado.

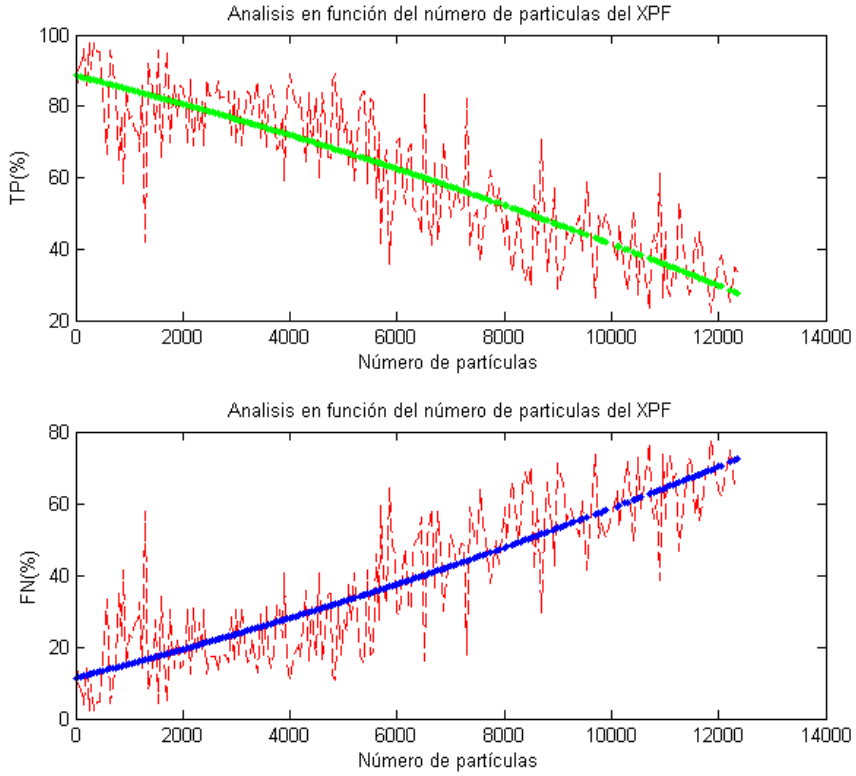

Figura 4.4: Análisis paramétrico en función de N_{PARTIC} evaluando TP(%) y FN(%).

En la figura 4.4 se muestran en la parte superior en rojo los valores del TPR obtenidos, y en verde

los valores de TPR obtenidos tras una interpolación cuadrática con 3 coeficientes. En la gráfica inferior se observa en azul los valores de FNR obtenidos tras una interpolación cuadrática con 3 coeficientes. El hecho de que los datos del TPR y FNR obtenidos (rojo) cuenten con una gran cantidad de ruido se podría reducir, dado que en el caso de realizar el estudio con una mayor cantidad de secuencias se obtendría un resultado mas estable.

De la figura 4.4 se observa que se produce un empeoramiento del funcionamiento del filtro de partículas con el aumento del numero de partículas totales, N_{PARTIC}, también observamos que evitando los valores con partículas sumamente bajos que puedan entorpecer la reinicialización del filtro de partículas extendido, entorno a $N_{PARTIC} = 2000$ partículas se observan los mejores valores para TPR, y por tanto este es el valor seleccionado como óptimo para el sistema implementado.

4.4 Comparación de resultados experimentales

En esta sección se muestran los resultados obtenidos tras el análisis de diversas secuencias por el sistema implementado, frente al mismo sin la etapa de seguimiento. En este caso al igual que en las secciones anteriores se presentan diversas tasas para determinar la efectividad y robustez del sistema implementado. Aun así, cabe destacar que una de las principales ventajas del sistema implementado es la posibilidad de conteo que ofrece, realizando un seguimiento de las personas por el entorno y proporcionando información de su trayectoria y velocidad entre otros, características de gran importancia para los sistemas de seguridad actuales.

A continuación se presentan las tablas 4.1 y 4.2 donde se observan los resultados obtenidos para el sistema implementado con y sin el sistema de clasificación. Los resultados se clasifican en función del numero de personas presentes en el entorno, como se puede observar.

Características de la detección							
Tipo de secuencia	Nºframes	TP	TN	FP	FN	TP (%)	FN (%)
SECUENCIAS DE UNA PERSONA	4550	4484	0	39	66	98.55 %	1.45 %
SECUENCIAS DE DOS PERSONAS	934	908	0	0	26	97.21 %	2.79 %
SECUENCIAS DE MAS DE DOS PERSONAS	8604	8470	0	4	134	98.44 %	1.56 %

Tabla 4.1: Resultados comparativos de la etapa de seguimiento sin la etapa de seguimiento.

Características de la detección							
Tipo de secuencia	Nºframes	TP	TN	FP	FN	TP (%)	FN (%)
SECUENCIAS DE UNA PERSONA	4625	4558	0	24	67	98.55 %	1.45 %
SECUENCIAS DE DOS PERSONAS	952	937	0	2	15	98.43 %	1.57 %
SECUENCIAS DE MAS DE DOS PERSONAS	8606	8286	0	7	320	96.28 %	3.72 %

Tabla 4.2: Resultados comparativos de la etapa de seguimiento con la etapa de seguimiento.

Como se observa en las tablas 4.1 y 4.2, los resultados obtenidos para las secuencias con una persona son muy similares, pero en el caso de las secuencias de dos personas, donde sin la etapa de seguimiento se obtenían los peores resultados, se han mejorado significativamente. Este hecho se debe a que en esta secuencias los dos individuos se mueven por el entorno muy juntos y por ello se pueden dar oclusiones entre ellos, gracias al uso del sistema de seguimiento se ha permitido la recuperación de la persona durante tales oclusiones. En el caso de las secuencias de mas de dos personas no se han mejorado los resultados, en estas personas se contaba con entornos con grupos de cuatro a ocho personas circulando libremente.

Aunque de forma general los valores obtenidos son muy buenos ya que el indice de que la persona clasificada sea persona ronda el 98.5 % para secuencias de una única persona, el 98.4 % para secuencias de dos personas y el 96.3 % para secuencias de mas de dos personas, uno de los objetivos del sistema radica en el conteo y seguimiento de las personas detectadas, en la figura 4.5 se observa una secuencia con varias personas, donde se puede observar la trayectoria de la persona a lo largo de los últimos frames que esta presente en la imagen, donde se ve identificada en su recorrido. Cabe destacar que sin la etapa de seguimiento implementado el sistema puede clasificar, pero no realizar un conteo ni seguimiento de las personas durante las secuencias.

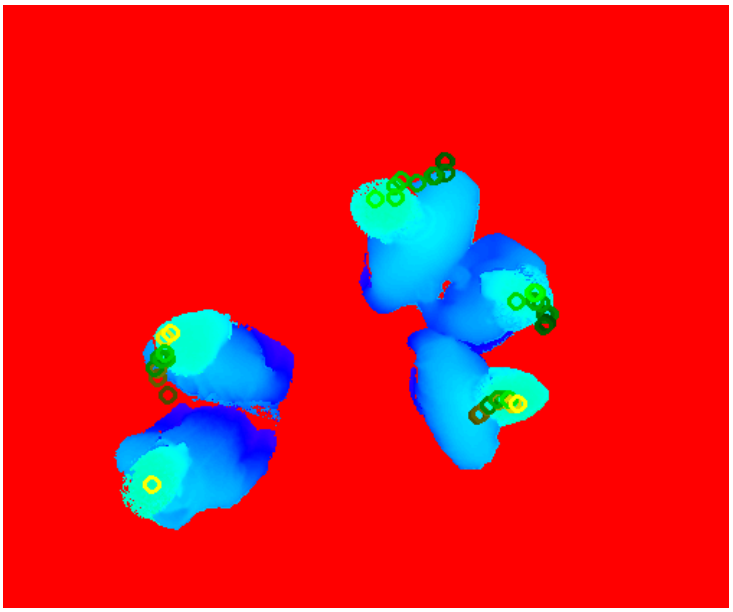

Figura 4.5: Secuencia de un frame con 5 personas del sistema con la etapa de seguimiento implementada.

4.5 Conclusiones

En esta sección se han mostrado los resultados obtenidos por el sistema implementado, comparando su efecto con el sistema sin la etapa de clasificación propuesta. En el presente trabajo se ha buscado obtener resultados lo mas reales posible, por lo que se han empleado secuencias de imágenes que reflejen situaciones realistas en entornos no controlados, como es el caso de secuencias con individuos agrupados o las de dos personas muy juntas, obteniendo buenos resultados para la detección de las personas. Por otro lado se cumple uno de los objetivos mas importantes de estos sistemas, el conteo y seguimiento del sistema, hecho que sin la ayuda del sistema de seguimiento resulta inviable.

Capítulo 5

Conclusiones y líneas futuras

En este apartado se resumen las conclusiones obtenidas y se proponen futuras líneas de investigación que se deriven del trabajo.

5.1 Conclusiones

En el presente trabajo se ha implementado un sistema robusto para la detección, seguimiento y conteo de personas. Para ello, se emplea únicamente información de profundidad adquirida empleando una cámara Kinect II ubicada en posición cenital.

Para la extracción de vectores de características se han evaluado dos alternativas diferentes llegando a la conclusión de que los descriptores de características propuestos en [7] no son válidos para la aplicación planteada en este trabajo. Por ese motivo, en el sistema implementado se emplean los descriptores de profundidad específicos para la detección de personas a partir de información de una cámara ToF cenital. El clasificador elegido está basado PCA e incluye tres clases diferentes: persona, persona con sombrero y no persona.

El seguimiento de múltiples personas se lleva a cabo empleando un XPFCP, elegido debido a su carácter multimodal que permite el seguimiento de un número variable de personas con un único filtro.

Para evaluar la solución implementada se han realizado diversas pruebas experimentales para obtener las tasas de verdaderos positivos y falsos positivos y comparar la validez del sistema y el impacto de incorporar la etapa de seguimiento. En las pruebas realizadas se ha comprobado que los resultados son satisfactorios alcanzo tasas de detección del 98.5 % para secuencias de una única persona, 98.4 % para secuencias de dos personas y 96.3 % para secuencias de más de dos personas. Cabe destacar que en el caso de una y dos personas se han conseguido mejoras al incorporar el filtro de partículas.

El caso en el que se ha obtenido una mayor mejora, es el de secuencias con dos personas muy agrupadas, ello se debe a que el sistema de seguimiento ha permitido evitar la influencia de las oclusiones entre las personas. Además, el sistema de seguimiento permite realizar el conteo de las personas, así como conocer sus trayectorias en cada momento.

En resumen, se puede concluir que se han alcanzado con éxito todos los objetivos planteados al inicio de este trabajo

5.2 Líneas futuras

A continuación se describen futuras mejoras para el sistema implementando, las cuales aporten beneficios en el ámbito del desarrollo de este trabajo.

- **Empleo de otros descriptores de imágenes en profundidad:** el estudio de otros descriptores de profundidad puede dar lugar a el uso de otras características de las personas del entorno, ello posibilitaría una mejora en la detección de las personas del entorno.
- **Clasificación de otros elementos del entorno:** la detección de diferentes elementos proporcionaría información del comportamientos de las personas con los ítems presentes en el entorno. Por ejemplo la detección de una mochila depositada en el entorno permitiría conocer quien la ha colocado y que trayectoria a seguido hasta tal lugar.
- **Empleo de cámaras de tiempo de vuelo con mejores características:** en el mercado actual se pueden encontrar cámaras con mejor resolución que la Kinect II, como las cámaras desarrolladas por *Basler*, [20]. También se cuenta con cámaras cuya distancia máxima mesurable sin error permitirían su colocación en posiciones mas altas para tener un mayor campo de visión(FOV, Field of view). Gracias a estas cámaras, las imágenes obtenidas tendrían mayor calidad y por tanto se podrían extraer mayor cantidad de características.
- **Uso de cámaras térmicas:** las cámaras térmicas se emplean cada vez mas en la detección de personas, actualmente se esta incrementando su uso debido a el descenso de su precisión y la mejora de sus características, en los trabajos desarrollados en [21] y [22], se puede observar un ejemplo de su uso en este ámbito. El uso de estas cámaras podría favorecer su empleo en situaciones de baja luminosidad como las zonas fronterizas.

Bibliografía

[1] M. Hansard, S. Lee, O. Choi, and R. Horaud, *Time of Flight Cameras: Principles, Methods, and Applications*, ser. SpringerBriefs in Computer Science. Springer, Oct. 2012. [Online]. Available: https://hal.inria.fr/hal-00725654

[2] J. Sell and P. O'Connor, "The xbox one system on a chip and kinect sensor," *Micro, IEEE*, vol. 34, no. 2, pp. 44–53, Mar 2014.

[3] L. C. B. Héctor Valdés-González, "Detección de perdidas en tuberias de agua: propuesta basada en un banco de filtros," *Ingeniare. Revista chilena de ingenieria*, vol. 17, pp. 375 – 385, 12 2009. [Online]. Available: http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-33052009000300011&nrm=iso

[4] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE MultiMedia*, vol. 19, no. 2, pp. 4–10, Apr. 2012. [Online]. Available: http://dx.doi.org/10.1109/MMUL.2012.24

[5] R. G. Jimemez, "Detección y conteo de personas, a partir de mapas de profundidad cenitales capturados con cámaras tof." Ph.D. dissertation, Universidad de Alcalá, 2015.

[6] M.-K. Hu, "Visual pattern recognition by moment invariants, computer methods in image analysis," *IRE Transactions on Information Theory*, vol. 8, 1962.

[7] L. Bo, X. Ren, and D. Fox, "Depth kernel descriptors for object recognition," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2011, pp. 821–826.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," 2003.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, June 2005, pp. 886–893 vol. 1.

[10] I. Jolliffe, *Principal Component Analysis*. Springer Verlag, 1986.

[11] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*, 1995, pp. 273–297.

[12] H. Drucker, D. Wu, and V. Vapnik, "Support vector machines for Spam categorization," *IEEE-NN*, vol. 10, no. 5, pp. 1048–1054, 1999.

[13] E. J. C. Suarez, "Tutorial sobre maquinas de vectores de soporte," 2013.

[14] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[15] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psych.*, vol. 24, 1933.

[16] P. Meinicke, T. Lingner, A. Kaever, K. Feussner, C. Göbel, I. Feussner, P. Karlovsky, and B. Morgenstern, "Metabolite-based clustering and visualization of mass spectrometry data using one-dimensional self-organizing maps," *Algorithms for Molecular Biology*, vol. 3, no. 1, pp. 1–18, 2008. [Online]. Available: http://dx.doi.org/10.1186/1748-7188-3-9

[17] C. L. Gutiérrez, "Segmentación y posicionamiento 3d de robots móviles en espacios inteligentes mediante redes de cámaras fijas." Ph.D. dissertation, Universidad de Alcalá, 2010.

[18] D. Luenberger, "Observers for multivariable systems," in *IEEE Transactions on Automatic Control 11*, 1966, pp. 190–197.

[19] J. García, A. Gardel, I. Bravo, J. Luis Lázaro, M. Martínez, and D. Rodríguez, "Detección y seguimiento de personas basado en estereovisión y filtro de kalman," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 9, no. 4, Apr. 2012.

[20] "Página del fabricante basler," http://www.baslerweb.com/ [Último acceso 1/septiemre/2016].

[21] A. Königs and D. Schulz, "Evaluation of thermal imaging for people detection in outdoor scenarios," in *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*, Nov 2012, pp. 1–6.

[22] J. Portmann, S. Lynen, M. Chli, and R. Siegwart, "People detection and tracking from aerial thermal views," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 1794–1800.

[23] "Información sobre gnu/linux en wikipedia," http://es.wikipedia.org/wiki/GNU/Linux [Último acceso 1/enero/2016].

[24] "Página de la aplicación emacs," http://savannah.gnu.org/projects/emacs/ [Último acceso 1/enero/2016].

[25] "Página de la aplicación kdevelop," http://www.kdevelop.org [Último acceso 1/noviembre/2013].

[26] L. Lamport, *LaTeX: A Document Preparation System, 2nd edition.* Addison Wesley Professional, 1994.

[27] "Página de la aplicación cvs," http://savannah.nongnu.org/projects/cvs/ [Último acceso 1/enero/2016].

[28] "Página de la aplicación gcc," http://savannah.gnu.org/projects/gcc/ [Último acceso 1/enero/2016].

[29] "Página de la aplicación make," http://savannah.gnu.org/projects/make/ [Último acceso 1/enero/2016].

[30] "Página de la aplicación opencv," http://opencv.org/ [Último acceso 5/septiembre/2016].

Apéndice A

Manual de usuario

A.1 Introducción

En el presente manual de usuario se exponen las características del sistema software desarrollado en este proyecto. Todas las funciones implementadas, han sido programadas en C/C++ haciendo uso de las librerias OpenCV 2.4.9. Por lo cual, para el correcto funcionamiento del sistema resulta imprescindible la previa instalación de tales librerías, así como un sistema compatible con los requisitos de las mismas, en este caso Linux Ubuntu 14.04.1 LTS o superior.

A.2 Manual

El manual aquí descrito esta basado en el sistema de detección y conteo de personas implementado, prestando especial atención al sistema de clasificación.

Dentro del código realizado del XPF se emplean diversas estructuras cuyo contenido es de suma importancia para entender el proceso, una de ellas es el *XpfCluster*, dentro de la cual se encuentran:

- *identify*: contiene el identificador asociado a cada cluster.
- *candidate*: representa el valor de un contador que crece o decrece en función de la permanencia o no de la persona en el entorno.
- *validated*: cuando *candidate* alcanza un valor determinado que indica que la persona ya se encuentra en el entorno con seguridad *validated* se encontrará a 1 mientras que en su defecto se encuentra a 0.
- *nMembers*: cada cluster tiene asociados un número determinado de partículas
- *centroid*: el cluster se encuentra posicionado por su centroide, su coordenada x, y, altura y velocidades en cada eje en el caso de ser necesario.
- *members*: para las *nMembers* partículas del cluster se almacenan sus respectivas coordenadas.

Los ficheros empleados por el sistema son 4, sus extensiones son *.gt*, *.z16*, *.result* y *.xpf*

Las imágenes de profundidad obtenidas de la cámara *Kinect II* se almacenan en ficheros de extensión *.z16* en los cuales únicamente se encuentran procesadas las imágenes. Para las secuencias de entrenamiento se cuenta con un fichero asociado a cada secuencia, este archivo contará con extensión *.gt* y en el encontraremos la siguiente información:

- Número del frame de la secuencia.
- Número de personas que se encuentran dentro del entorno en cada frame.
- Para cada una del las personas se cuenta con 6 puntos característicos con 2 coordenadas,x e y, 3 de los puntos indican la posición de la cabeza, mientras que los 3 restantes indican la posición y orientación de los hombros. Gracias a esta información en etapas posteriores podemos determinar la región de interés que queremos evaluar en el caso de las secuencias de entrenamiento.

En el caso de los ficheros de extensión *.result*, en su contenido se almacenan los resultados obtenidos para la etapa de clasificación. Dentro de ellos encontramos la siguiente información referente a las personas clasificadas.

- Número de frame de la secuencia.
- Indice de clasificación: el cual toma los valores 0 o 1, 0 para la clasificación dentro de la clase persona y 1 en el caso de que la persona se clasifique dentro de la clase persona con accesorio.
- Posición de la persona dentro del entorno, indicando 3 valores:
  - Coordenada pixélica u.
  - Coordenada pixélica v.
  - Altura (cm).
- Error de recuperación de la etapa de clasificación PCA.

El sistema implementado genera ficheros de extension *.xpf*, en ellos encontramos los valores de las personas detectadas y seguidas por el sistema, a continuación se detalla su estructura.

- Número de frame de la secuencia.
- Indice de identificación: el cual toma los valores desde -1 hasta $N_{personas}$, donde $N_{personas}$ representa las posibles personas presentes en el entorno y el -1 que la persona no clasificada no se contemple en el filtro de partículas implementado.
- Posición de la persona dentro del entorno, indicando 3 valores:
  - Coordenada pixélica u.
  - Coordenada pixélica v.
  - Altura (cm).
- Indice representativo de la relación entre detección y seguimiento, tomando los valores 0, 0.5, 1 y 2. Donde 0 indica que la persona detectada se encuentra en la etapa de seguimiento, 0.5 que la persona solo se encuentra en la etapa de seguimiento pero no se detecto en la clasificación, 1 que se encuentra en la etapa de seguimiento y clasificada pero se asociaron por distancia, y 2 en el caso de solo encontrarse detectada pero no en la etapa de clasificación.

A.3 Ejecución del software

Los ficheros fuentes necesario para el correcto funcionamiento del sistema implementado se encuentran en el directorio *tofCountingExperimentAlvaro+XPF* dentro de la carpeta de librerías de grupo de investigación GEINTRA.

Una vez situados en tal directorio se ha de compilar el código, para lo cual en el terminal de comandos de Linux se ejecuta el comando *make*, y tras ello se procede a ejecutar el programa, para lo cual se ejecuta de el comando *./tofCountingXPF* al cual se han de adjuntar 4 argumentos:

- Primer argumento: se trata de un fichero de extensión *.list* que ha de contener el directorio y nombre de las secuencia que se quiera ejecutar (por ejemplo: */tofCountingScoring/seq-P05-M04-A0001-G03-C00-S0030/seq-P05-M04-A0001-G03-C00-S0030.z16*)
- Segundo argumento: en este caso se ha de indicar el parámetro de la distancia máxima considerada para la asociación de clusters, $Distancia_{MAX}$.
- Tercer argumento: el parámetro que se ha de indicar representa el porcentaje de partículas que se añaden al filtro de partículas extendido implementado en cada iteración en la que se encuentre activo, $N(\%)_{PARTIC_{NEW}}$.
- Cuarto argumento: se trata de el número de partículas del filtro de partículas extendido implementado, N_{PARTIC}.

En el directorio donde se encuentra el el fichero de extensión *.z16* que se indica en el primer argumento de la ejecución, e decir de extensión *.list*, se ha de encontrar también el fichero de extensión *.result* correspondiente a la misma secuencia. Este fichero corresponde a los resultados intermedios obtenidos tras la etapa de clasificación, en la imagen A.1 se muestra un ejemplo del fichero *.result*.

```
413   0 252 111 180 0.066   0 118 87 205 0.415
414   0 253 127 181 0.181   0 118 86 205 0.401
415   0 247 134 181 0.119   0 114 92 207 0.245
416   0 248 146 182 0.247
417   0 247 151 182 0.231   0 119 102 206 0.150
418   0 245 156 182 0.118   0 118 110 205 0.334
419   0 247 154 183 0.145   0 119 117 205 0.175
420   0 245 175 184 0.169   0 121 110 205 0.171
421   0 248 184 185 0.246   0 133 117 204 0.129
422   0 252 182 184 0.186   0 124 137 203 0.073
423   0 250 192 185 0.183   0 118 144 203 0.580
424   0 250 208 185 0.188   0 129 144 203 0.125
425   0 253 220 185 0.224   0 131 167 203 0.126
426   0 244 221 184 0.189   0 127 182 202 0.253
427   0 249 232 184 0.238   0 136 188 203 0.130
428   0 251 233 183 0.193   0 140 199 204 0.200
429   0 251 250 183 0.236   0 134 207 203 0.282
430   0 253 254 182 0.226   0 140 218 204 0.220
431   0 263 261 182 0.230   0 143 227 204 0.105
432   0 257 279 181 0.237   0 144 234 204 0.193
433   0 260 283 181 0.259   0 149 242 204 0.161
434   0 264 281 180 0.114   0 131 259 205 0.218
435   0 263 301 181 0.184   0 141 261 205 0.142
436   0 264 297 181 0.088   0 139 271 205 0.143
437   0 250 318 182 0.230   0 141 283 205 0.106
438   0 267 315 182 0.199   0 130 273 204 0.260
439   0 265 331 183 0.280   0 132 276 204 0.260
440   0 131 287 204 0.293
441   0 132 292 203 0.209
442   0 132 320 203 0.199
```

Figura A.1: Ejemplo del fichero de extensión *.result*.

Tras la ejecución del programa *./tofCountingXPF*, se obtiene en el mismo directorio del mismo ejecutable el fichero de extensión *.xpf* que contiene los resultados del sistema. En la imagen A.2 se muestra un ejemplo del fichero *.xpf*.

```
2  369  150  184    1
3  246  149  182    0     2  374  146  184    0
3  248  142  182    1     2  369  143  184    0     4  315  338  185    1
1  239  143  182    0     2  378  134  184    1
3  239  148  182    0     2  368  135  184    0
2  371  130  184    0     4  313  334  185    1
1  237  146  182    0     2  365  126  184    0     4  323  333  186    1
1  246  146  182    1     2  355  130  184    0     4  311  335  186    0
1  234  140  182    1     2  364  118  184    0     4  319  320  185    1
1  230  149  183    1     4  317  316  184    0     2  367  122  185    0
1  235  142  182    1     4  314  319  184    0     2  364  118  185    0
1  234  150  182    0     4  322  313  184    1     2  362  117  185    0
1  230  153  183    0     4  320  315  184    0     2  355  115  185    0
1  227  149  183    0     2  352  112  184    0     4  328  307  184    1
1  218  165  183    1     2  354  114  184    0     4  332  306  184    1
1  217  157  182    0     2  352  113  184    0     4  339  296  185    1
1  217  160  182    0     2  347  112  184    0     4  337  297  185    0
1  208  157  183    0     2  356  109  184    0     4  339  288  185    1
5  438  218  182    1     1  211  170  183    0     2  346  113  184    0     4  341  294  185    0
1  206  166  183    0     5  440  215  183    1     3  347  117  185    0     4  340  285  185    0
1  209  173  183    0     5  441  213  183    1     2  339  116  184    0     4  348  291  186    1
1  203  171  182    0     5  437  203  182    1     2  338  116  184    0     4  350  279  185    1
1  203  169  182    0     5  434  201  182    1     2  336  117  184    0     4  348  283  186    0
1  198  163  182    0     5  432  196  182    1     3  342  117  185    0     4  353  272  185    1
5  432  192  182    1     1  197  179  183    0     3  343  119  185    0     4  356  272  186    1
1  202  175  182    0     5  430  190  182    1     3  341  117  185    0     4  353  267  185    0
5  428  187  182    0     3  342  112  185    0     4  357  268  186    0
5  423  181  181    0     1  198  183  183    0     3  334  119  185    0     4  363  270  186    1
5  424  176  181    0     1  197  183  183    0     2  339  117  185    0     4  355  259  185    0
5  423  172  181    0     1  191  179  183    0     2  337  112  185    0     4  356  260  185    0
5  431  166  181    1     2  337  112  185    0     4  359  254  185    0
5  424  157  180    1     1  184  182  183    0     2  338  114  185    0     4  363  257  186    0
5  424  153  180    0     1  183  192  183    0     3  322  116  185    0     4  355  251  185    0
5  424  153  180    0     3  319  118  185    0     4  354  244  185    0
```

Figura A.2: Ejemplo del fichero de extensión *.xpf*.

Apéndice B

Herramientas y recursos

Las herramientas necesarias para la elaboración del proyecto han sido:

- PC compatible
- Sistema operativo GNU/Linux [23]
- Entorno de desarrollo Emacs [24]
- Entorno de desarrollo KDevelop [25]
- Procesador de textos LaTeX[26]
- Control de versiones CVS [27]
- Compilador C/C++ gcc [28]
- Gestor de compilaciones make [29]
- Librerías OpenCV [30]

Apéndice C

Pliego de condiciones

Para la correcta utilización del sistema desarrollado en este trabajo se debe disponer de un hardware y un software que cumpla unos requisitos mínimos.

C.1 Requisitos de Hardware

- Procesador de 32/64 bits
- 2GB de memoria RAM o superior
- Al menos 400MB de memoria libres en el disco duro para funciones y datos.
- Al menos 10GB de memoria libres en el disco duro para la base de datos etiquetada y los ficheros generados (archivos de extensión *.gt, .result, .z16 y .xpf*)
- Camara Kinect II o similar

C.2 Requisitos de Software

- Sistema operativo Linux Ubuntu 14.04.1 LTS
- Libreria OpenCV 2.4.9
- Al menos 400MB de memoria libres en el disco duro para funciones y datos.
- Compilador GNU GCC

Apéndice D

Presupuesto

D.1 Costes de equipamiento

- Equipamiento Hardware empleado

Concepto	Cantidad	Coste Unitario	Subtotal
PC Asus I7 8GB de RAM	1	700€	700€
Kinect II	1	200€	200€
Coste total HW			900€

Tabla D.1: Costes del equipamiento Hardware empleado.

- Equipamiento Software empleado

Concepto	Cantidad	Coste Unitario	Subtotal
Ubuntu 14.04 LTS	1	0€	0€
Libreria OpenCV 2.4.12	1	0€	0€
Software LaTeX	1	0€	0€
Coste total SW			0€

Tabla D.2: Costes del equipamiento Software empleado.

D.2 Costes de mano de obra

Concepto	Cantidad	Coste Unitario	Subtotal
Desarrollo SW	200	65€/ hora	13000€
Mecanografiado del documento	100	15€/ hora	1500€
Coste total de la mano de obra			14500€

Tabla D.3: Costes de la mano de obra empleada.

Concepto	Subtotal
Equipamiento Hardware	900€
Recursos Softhare	0€
Mano de obra	14500€
Coste total del presupuesto	15400€

Tabla D.4: Coste total del presupuesto.

D.3 Coste total del presupuesto

El importe total del presupuesto asciende a la cantidad de: QUINCEMIL CUATROCIENTOS EUROS

En Alcalá de Henares a___de_____de 2016

Álvaro Fernández Rincón.

Graduado en Ingenieria en Electronica y Automatica Industrial.

Universidad de Alcalá
Escuela Politécnica Superior

ESCUELA POLITECNICA
SUPERIOR

Universidad
de Alcalá