Universidad de Alcalá
Escuela Politécnica Superior

Grado en Ingeniería Electrónica de Comunicaciones

Trabajo Fin de Grado

Clasificación de accesorios a partir de información de profundidad.

Autor: Antonio Carlos Cob Parro

Tutor: Cristina Losada Gutiérrez

2018

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Electrónica de Comunicaciones

Trabajo Fin de Grado

Clasificación de accesorios a partir de información de profundidad.

Autor: Antonio Carlos Cob Parro

Tutor: Cristina Losada Gutiérrez

Tribunal:

Presidente: Daniel Pizarro Pérez

Vocal 1º: Marta Marrón Romera

Vocal 2º: Cristina Losada Gutiérrez

Calificación: .....

Fecha: .....

Agradecimientos

Gracias por todas aquellas manos que no me han dejado caer. Gracias a mis padres Paloma y Antonio. Gracias a mi pareja Andrea y gracias a todos aquellas personas que desde la sombra me apoyaron.

Resumen

El objetivo de este trabajo de fin de grado (TFG) es la identificación robusta de complementos a partir de imágenes de profundidad (2.5D). Dichas imágenes serán adquiridas de una cámara Kinect II ubicada en una posición cenital. Los complementos evaluados en este caso son gorras y distintos tipos de sombreros (grandes, pequeños y medianos), que la solución propuesta debe ser capaz de identificar. La solución propuesta extrae un conjunto de descriptores por cada persona previamente detectada en la escena que, posteriormente, son clasificados utilizando la técnica PCA (Análisis de Componentes Principales), comparándolos con las distintas clases previamente entrenadas. El sistema desarrollado se ha evaluado realizando diferentes pruebas experimentales sobre secuencias de profundidad reales, obteniendo resultados satisfactorios. En concreto, se han obtenido tasas de acierto del 98 % para el caso más sencillo (clasificación binaria) y superiores al 85 % en los casos más complejos (cuatro o cinco clases).

Palabras clave: PCA, Análisis de Componentes Principales, Clasificación, Distancia Euclídea, Distancia de Mahalanobbis.

Abstract

The aim of this final degree thesis is the robust identification of headgear accesories from depth images (2.5D) acquired using a Kinect II camera located in a zenithal position. The accesories evaluated in this work are caps and different types of hats (large, small and medium). The proposed solution must be able to identify complements of each class. The proposed solution extracts a set of descriptors for each person previously detected in the scene, which are then classified using the PCA (Principal Component Analysis) technique, comparing them with the different classes previously trained. The developed system has been evaluated by carrying out different experimental tests on real depth sequences, obtaining satisfactory results. Specifically, success rates of 98 % have been obtained for the simplest case (binary classification) and higher than 85 % in the most complex cases (four or five classes).

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

La detección de personas, el análisis de su comportamiento en aplicaciones de vídeo-vigilancia y clasificación de objetos son temas ampliamente estudiados por la comunidad científica en las áreas de la visión e inteligencia artificial, debido a sus múltiples aplicaciones en diferentes tareas como la video-vigilancia, control de aforos, seguridad, análisis de comportamientos, etc.

Con la evolución de la tecnología y las telecomunicaciones, estos temas se hacen relevantes, surgiendo numerosos trabajos que tratan de resolverlos.

En este contexto en el presente Trabajo de Fin de Grado (TFG) se aborda el diseño, implementación y evaluación de un sistema de clasificación de complementos utilizados por usuarios (accesorios para la cabeza como gorras, sombreros, etc.) de forma robusta, y en tiempo real. Todo ello a partir de la información de distancia proporcionada por un sensor de profundidad basado en tiempo de vuelo (ToF) ubicado en posición cenital. En concreto se emplea un sensor Kinect II [6, 7], el cual permite obtener tanto información de profundidad, como imágenes en color RGB (Red, Green Blue), imagen de amplitud infrarroja e información de audio.

En el caso de este TFG únicamente se emplea la información de profundidad ya que en muchas de las posibles aplicaciones existen limitaciones relacionadas con la preservación de la privacidad de los usuarios. Este tema genera un debate muy extenso sobre el uso correcto de la información recogida para su posterior tratamiento. En el caso de la información de profundidad, no es posible reconocer la identidad de los usuarios, por lo que permite cumplir las restricciones relacionadas con la privacidad.

Para la realización de este TFG se han tomado como punto de partida los trabajos previos realizados por miembros del grupo de investigación GEINTRA [8](Grupo de Ingeniería Electrónica Aplicada a Espacios Inteligentes y Transporte) para la detección y seguimiento de personas a partir de información de profundidad [9–11] incorporando al detector las etapas de extracción de características y la clasificación robusta de complementos

A continuación se describe con mayor detalle el objetivo concreto del TFG, la metodología empleada y la solución propuesta. Además, se detalla la organización de la presente memoria.

1.1. Objetivos

Como ya se ha comentado, el objetivo general de este TFG es la clasificación de complementos a partir de la información de profundidad proporcionada por una cámara ToF ubicada en posición cenital. Debido a la ubicación del sensor, los complementos a clasificar incluyen aquellos que se llevan en la cabeza: gorras, sombreros, gorros, etc.

Este objetivo general se divide en objetivos más sencillos que se detallan a continuación:

1. Puesta en marcha del sistema de detección de personas previamente desarrollado en los trabajos [9, 10].
2. Extracción de descriptores a partir de la información de profundidad, dentro de la Región de Interés (ROI) alrededor de cada persona detectada.
3. Clasificación de descriptores para determinar el complemento o la ausencia de este.
4. Evaluación exhaustiva para la validación del sistema desarrollado
5. Documentación del trabajo realizado.

Para alcanzar estos objetivos, es necesario abordar diferentes tareas que se detallan en el siguiente apartado.

1.2. Metodología

Como se ha comentado anteriormente, el objetivo de este TFG es el diseño, implementación y evaluación de un sistema de detección y clasificación de accesorios completo y robusto. Para alcanzar este objetivo se han abordado las tareas que se presentan a continuación.

1. Estudio y puesta en marcha del código que ha sido utilizado para la correcta detección de individuos. En esta primera toma de contacto se aprende la estructura y el lenguaje empleado en el software.
2. Estudio y evaluación de diferentes métodos para la extracción y clasificación de descriptores de características.
3. Implementación de la algoritmia para la clasificación de complementos y evaluación experimental de cada etapa.
4. Evaluación exhaustiva del sistema propuesto y estudio de los resultados obtenidos. En caso necesario, mejora del software desarrollado para alcanzar los resultados deseados.
5. Documentación del trabajo desarrollado.

1.3. Solución Propuesta

La figura 1.1 muestra un diagrama de bloques general del sistema desarrollado. Como se puede observar, el proceso de clasificación consta de dos etapas bien diferenciadas, una etapa *off-line* en la que se realiza el entrenamiento de las diferentes clases y se ejecuta una una vez al arrancar el sistema, y otra *on-line* en la que se lleva a cabo la clasificación robusta de los complementos. Estas dos etapas tienen elementos en común. Como por ejemplo la detección de la ROI (Region of Interest), la extracción del mapa de profundidad y la extracción y normalización de características.

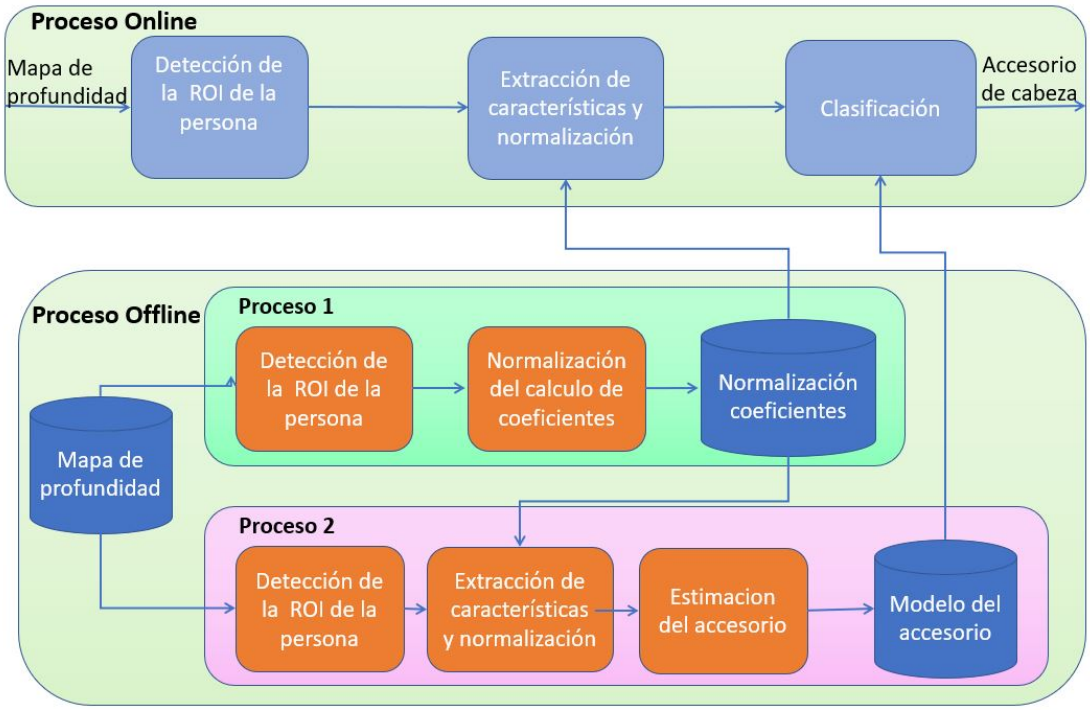


Figura 1.1: Diagrama de bloques con el funcionamiento global del sistema (basado en [1]).

- **Etapa Off-line**: se realiza la detección de la persona y la ROI asociada, para posteriormente realizar la extracción y normalización de los vectores de características. El proceso de normalización es necesario para que los vectores obtenidos sean invariantes a la altura de la persona. Finalmente, los vectores generados se emplean para definir las diferentes clases de complementos a clasificar. Este proceso solo se realiza una vez.

  - **Detección de la ROI de la persona**: dado que se trata de una fase de entrenamiento, se define la ROI utilizando el algoritmo implementado en [10]. Para ello se emplea un proceso en el cual se seleccionan ocho puntos característicos de la persona evaluada, cinco correspondientes a su cabeza y tres de sus hombros y nuca. Conforme a tales puntos se selecciona la región de interés a evaluar
  - **Cálculo de los coeficientes de normalización**: se calculan los coeficientes que son usados para que se lleve acabo la normalización del vector característica.

- **Normalización de coeficientes**: mediante el uso de los coeficientes calculados en el apartado anterior se lleva acabo la normalización del vector características.
- **Estimación de modelos de accesorios**: una vez obtenidas las ROI's, tras la extracción y normalización de los vectores características se obtienen los modelos de cada uno de los complementos a clasificar, que se emplearán en la etapa *on-line*.

- En el proceso **Online**: se repiten los pasos de detección de la persona, y la extracción y normalización de vectores de características (empleando los pesos calculados previamente). Tras ellos, se realiza la clasificación de los vectores para determinar los accesorios.

  - **Detección de la ROI de la persona**: la selección se lleva acabo en función de los máximos encontrados [10] y los valores de los píxeles alrededor del máximo central.
  - **Extracción y normalización de los vectores de características**: una vez obtenida la ROI de la persona se lleva acabo la extracción de sus características, para obtener un vector de características.
  - **Clasificación**: se realiza una separación y ordenación de los complementos a partir de los vectores extraídos en el paso anterior, y los modelos previamente calculados en la etapa *off-line*.

1.4. Estructura del documento

El TFG 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 el sistema de clasificación de complementos y el método para abordar con éxito el TFG.
2. Fundamentos Teóricos: en dicho capitulo se abordaran los diferentes conceptos teóricos que aborda este TFG, dando una explicación de cada uno de ellos.
3. Detección y clasificación de complementos: en este apartado se expondrá el como se ha llevado a cabo el TFG, explicando en detalle cada uno de los apartados de la imagen de la sección anterior 1.1.
4. Resultados experimentales: en este capitulo se expondrán de forma ordenada y clara los resultados obtenidos en este TFG. También en este capitulo se aclarara el por que de esos resultados y que sentido tienen.
5. Conclusiones y líneas futuras: en este capitulo se evaluaran los resultados obtenidos, explicando si son exitosos. Tambien se explicara el como mejorar y como puede ser utilizado este TFG en proyectos futuros.

Capítulo 2

Fundamentos teóricos

2.1. Introducción

Como se ha comentado en el capítulo 1 el objetivo de este TFG es la detección de manera robusta y fiable de complementos. Los complementos utilizados son elementos de vestimenta que se utilizan en la cabeza (gorras, sombreros, pamelas...), debido a la ubicación cenital del sensor.

Para la adquisición de la información se emplea una cámara de ToF (Kinect II),que proporciona información de profundidad (distancia a cada punto de la escena).

Para la realización de este trabajo se ha tomado como punto de partida el trabajo realizado en [9, 10] para la detección de personas a partir de información de profundidad. Al detector desarrollado en esos trabajos previos, se le ha incorporado la nueva etapa de clasificación de complementos implementada en este TFG.

A lo largo de este capítulo se presentan los conceptos teóricos necesarios para la realización de este TFG, incluyendo las diferentes etapas de las que consta el trabajo: la adquisición de la información de profundidad, la extracción de descriptores de características y la clasificación de dichos descriptores.

2.2. Adquisición de la información del entorno

En este apartado se describen algunos de los elementos más utilizados para la adquisición de información. La adquisición se puede realizar tanto con sensores como con cámaras.

En el mercado hay una gran variedad de sensores que recogen información de su entorno, algunos tipos son:

- **Sensores infrarrojos**: dispositivo optoelectrónico que puede medir la radiación electromagnética en el campo infrarrojo, figura 2.1.

Figura 2.1: Sensor infrarrojo.

- **Sensores de ultrasonidos**: mediante la emisión de ondas que tienen una frecuencia mayor o igual a 40Khz inaudibles al oido humano [12]. Se utilizan para medir distancias mediante el calculo del retardo de la señal emitida, figura 2.2.

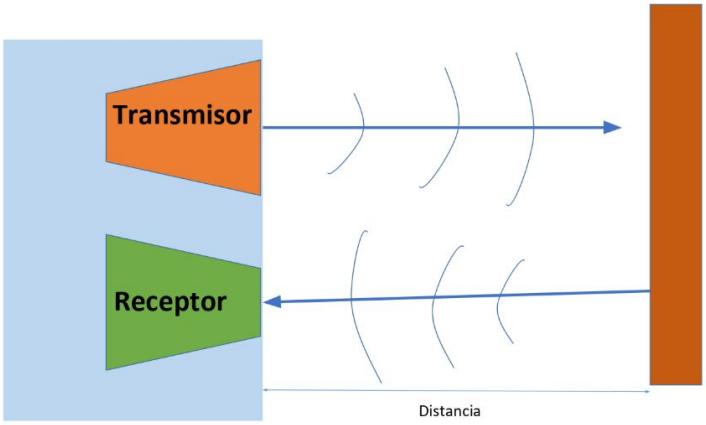


Figura 2.2: Esquema de funcionamiento de ultrasonidos.

Las cámaras también son elementos hardware pero con un nivel de complejidad mayor que el de los sensores. Muchas cámaras utilizan de muchos sensores para adquirir medidas y luego procesarlas.

- **Cámaras RGB**: este tipo de cámaras, son las más usuales. Son aquellas que recogen información en el espectro visible de la luz. La información recogida es tal y como el ojo humano es capaz de ver. Un ejemplo de una cámara RGB se puede ver en la figura 2.3

Figura 2.3: Cámara RGB [2].

- **Cámaras ToF**: ToF (del inglés time-of-flight) son un tipo de cámaras, figura 2.4 que estiman las distancias de los cuerpos mediante el cálculo del tiempo desde la emisión a la recepción de un haz de luz infrarrojos.

Figura 2.4: Cámara ToF [3].

Para este trabajo se han utilizado cámaras ToF por dos motivos. El primero es que se busca la privacidad de la identidad de las personas y como se ha comentado anteriormente. Este trabajo parte de un trabajo anterior de identificación de personas usando cámaras ToF. Por lo que ese es el segundo motivo. La cámara que se usa para este trabajo es la Kinect II 2.2.3.

2.2.1. Obtención de información de profundidad mediante cámaras de tiempo de vuelo

Las cámaras ToF se basan en la emisión y recepción de luz infrarroja. El emisor envía un haz de luz modulado con una frecuencia conocida (f_{mod}). Cuando esa luz rebota en un elemento opaco, una parte de la señal regresa a la cámara, siendo detectada por los sensores MOSFET incluidos. La diferencia de fase entre la señal emitida y la recibida permite estimar la distancia a la que se encuentra el objeto en el cual ha rebotado.

Este tipo de cámaras cuentan con dos elementos importantes. Se tratan del emisor y del receptor

- **Emisores**: para generar el haz de luz se pueden utilizar distintos elementos fotoemisores. El uso de LED o láseres en estado solido son los mas usados. La longitud de onda de estos emisores suele estar en torno a 850nm, por lo que no son visibles al ojo humano. La emisión del haz puede realizarse de forma continua o pulsada.

    - Modulación continua: se centra en iluminar de manera permanente el espacio deseado a medir. Esto hace que a la hora de rebotar el haz de luz en un objeto y volver al receptor se pueda observar un cambio de fase. Dicho cambio de fase es estudiado y proporciona información sobre la distancia a la que se encuentra el objeto, figura 2.5.

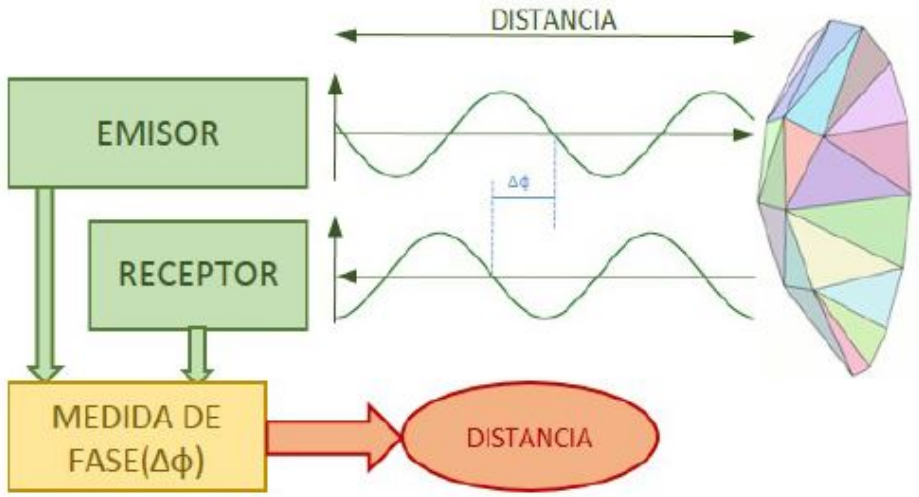


Figura 2.5: Modulación continua.

- Modulación pulsada: este método permite una medición directa gracias a un pulso. Además éste tipo de modulación sufre menos interferencias por la iluminación ambiental. Sin embargo requiere una gran precisión a la hora de medir los tiempos de salida y de entrada del pulso. Este problemas si lo juntamos con la dificultad de generar este tipo de pulsos con una frecuencia elevada hacen este tipo de modulación difícil de usar (figura 2.6).

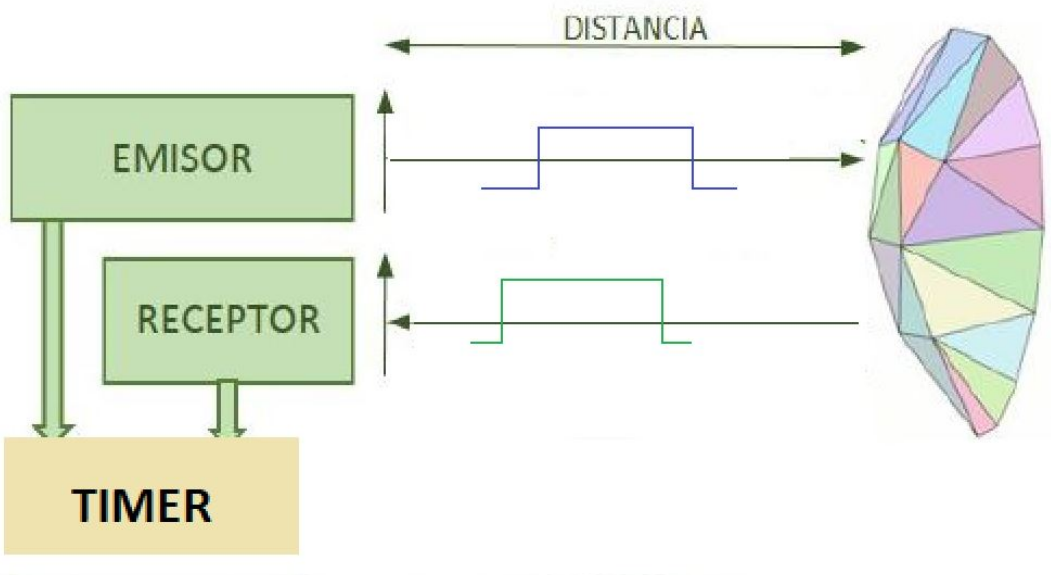


Figura 2.6: Modulación pulsada.

- **Receptores**: el receptor es la parte de la cámara que se encarga de recoger el haz de luz emitido.

Normalmente las cámaras ToF se basan en cuatro desplazamientos de fase. Cada desplazamiento corresponde a 90 grados. Estos desplazamientos se efectúan en cada uno de los píxeles. Ello implica que se usa un haz de luz durante un tiempo T, durante ese tiempo T se pueden distinguir los cuatro desplazamientos, figura 2.7.

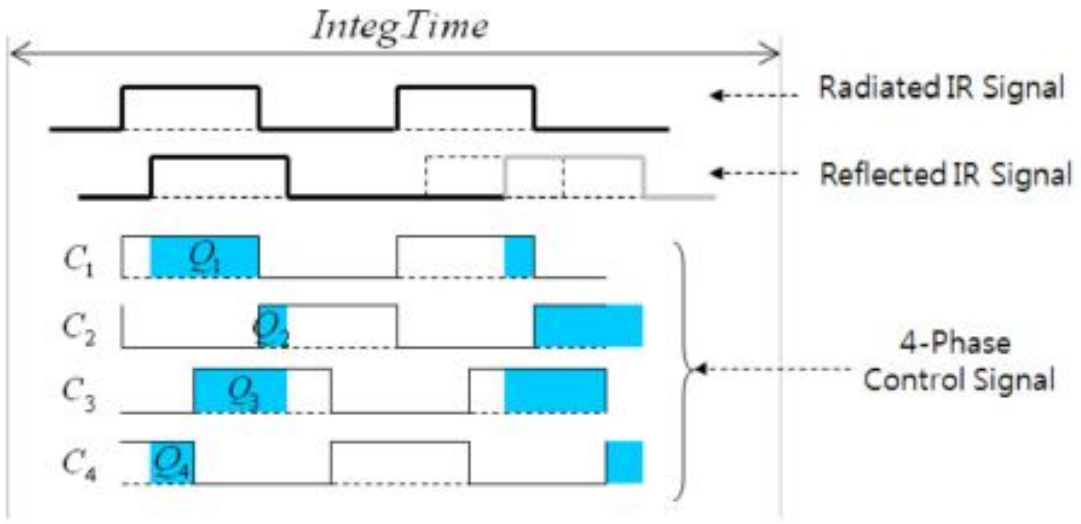


Figura 2.7: Funcionamiento de los transistores MOSFET durante los 4 períodos de recepción de el haz de luz, cuyo estudio se detalla en [4] de donde procede esta figura.

En la figura anterior 2.7 se observa el funcionamiento de la medida de la diferencia de fase del rayo emitido respecto al reflejado. Esto da lugar a cargas acumuladas durante las diferentes muestras de cada uno de los receptores: Q_1, Q_2, Q_3, Q_4. Para reducir los errores en la medida, las cargas se acumulan durante el denominado tiempo de integración antes de generar un valor de distancia. En la ecuación 2.1 se expresa de una manera matemática el cálculo del desfase a partir de las cargas acumuladas, siendo β la diferencia de fase entre el rayo emitido y el reflejado. En la ecuación 2.2 usando la β de la ecuación anterior, la velocidad de luz en el vacio $C = 3x10^8 m/s$ y la frecuencia de modulación del rayo emitido f_{mod} [13] se puede expresar de manera matemática la distancia, d.

$$\beta = \arctan(\frac{Q_3 - Q_4}{Q_1 - Q_2}) \tag{2.1}$$

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

A partir de la expresión anterior, puede obtenerse la distancia máxima que puede medirse sin ambigüedad para un sensor de modulación continua (d_{max}), mostrada en la ecuación 2.3. Debe haber una diferencia de fase entre 0 y 2π radianes entre los dos haces de luz. En el caso de que se intenten hacer medidas a una distancia superior a d_{max}, se producirán errores en la medida, ya que no es posible determinar la distancia real.

$$d_{max} = (\frac{c}{2f_{mod}}) \tag{2.3}$$

2.2.2. Fuentes de error en las medidas de profundidad de cámaras ToF

La medida de profundidad basada en tiempo de vuelo tiene múltiples fuentes de error que pueden provocar que la medida no sea correcta. A continuación se presentan los principales errores divididos en función de su naturaleza sistemática o aleatoria:

- **Errores sistemáticos**: estos errores aparecen 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 sistemáticos mas comunes destacan:
  - **Error *Wiggling***: este tipo de error se debe a la generación de imperfecta de luz modulada. Esto se puede observar cuando se genera una señal emitida sinusoidal perfecta y no lo es. Al recoger la información esa onda ya trae consigo un error.
  - **Variación en la amplitud**: es debida a la iluminación y la reflectividad generada por el entorno. En un entorno real no controlado es imposible mantenerlo constante.
  - **Ruido de disparo (*shotnoise*)**: los transistores de tipo MOSFET son elementos con poco numero de electrones. Cuando hay una pequeña variación en el numero de estos, se generan errores en la medida.
  - **Errores debido a la temperatura**: uno de los principales elementos de generación de error es la temperatura. En los materiales semiconductores se pueden ver afectados por este fenómeno.
- **Errores no sistemáticos**: se trata de aquellos errores no predecibles que se dan en las medidas de distancia.
  - ***Flying píxeles***: cuando dentro de la distancia de medida, se encuentran objetos o personas de los cuales se recoge información. Estas personas u objetos tienen un borde. En las medidas de los perímetros se observan medidas erróneas debidas a que en esos bordes se obtienen tanto medidas del objeto como del fondo durante el tiempo de integración, por lo que el resultado es un valor erróneo. Este tipo de error se ve mas acentuado cuando los objetos o personas están en movimiento produciendo el efecto conocido como artefactos de movimiento.
  - **Medición de la distancia máxima mensurable**: en las cámaras ToF cuanto mayor es la distancia entre ella y el objeto medido la amplitud del rayo infrarrojo recibido sera más débil, esto implica menor amplitud de la onda recibida. Por lo que este error depende de la amplitud con la que llega el rayo a la cámara, esta amplitud va a ser mayor dependiendo de la distancia a la que se encuentra el objeto y de la reflectividad del objeto en el que refleja. Por eso en los bordes este tipo de error sera mayor, ya que la distancia aumenta.

- **Ruido producido por la tonalidad**: este tipo de ruido aparece cuando la tonalidad de un objeto o persona es muy oscura. Esto es debido a que al usar infrarrojos la señal absorbida por ciertas tonalidades es bastante mayor. Por lo tanto no hay luz reflejada, eso implica una menor amplitud y con ello mas errores.

2.2.3. Sensor Kinect II

La Kinect II [6], mostrada en la figura 2.8, es una cámara desarrollada por la compañía Microsoft como periférico para la consola XBOX ONE, aunque permite también ser usada para situaciones fuera del ámbito de los videojuegos. Esta cámara genera tres salidas con tres formatos distintos: RGB, escala de grises e imágenes de profundidad.

Figura 2.8: Kinect II [5]

En la actualidad la Kinect II 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
- 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 figura 2.9 se pueden observar las distintas imágenes proporcionadas por la Kinect II: RGB, en escala de grises y de profundidad.

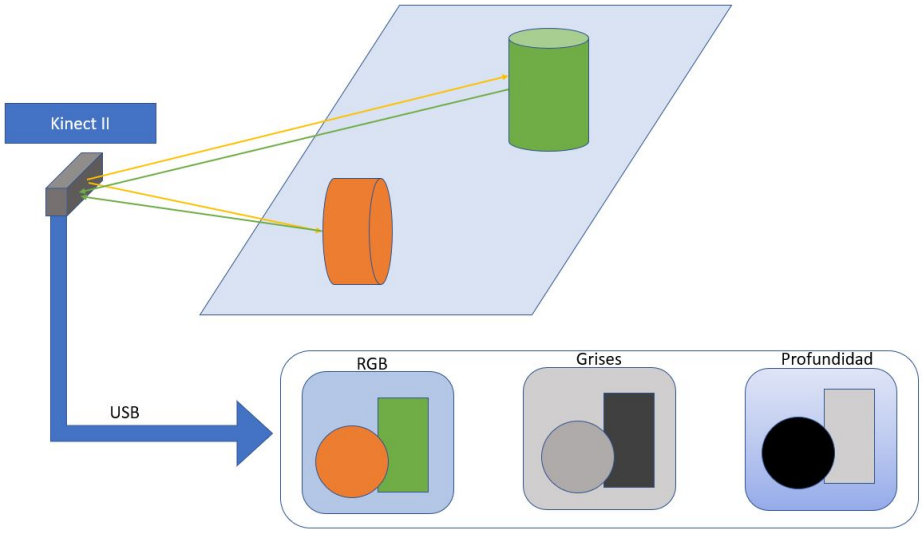


Figura 2.9: Distintas imágenes proporcionadas por la KinectII

En la figura 2.10 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*).

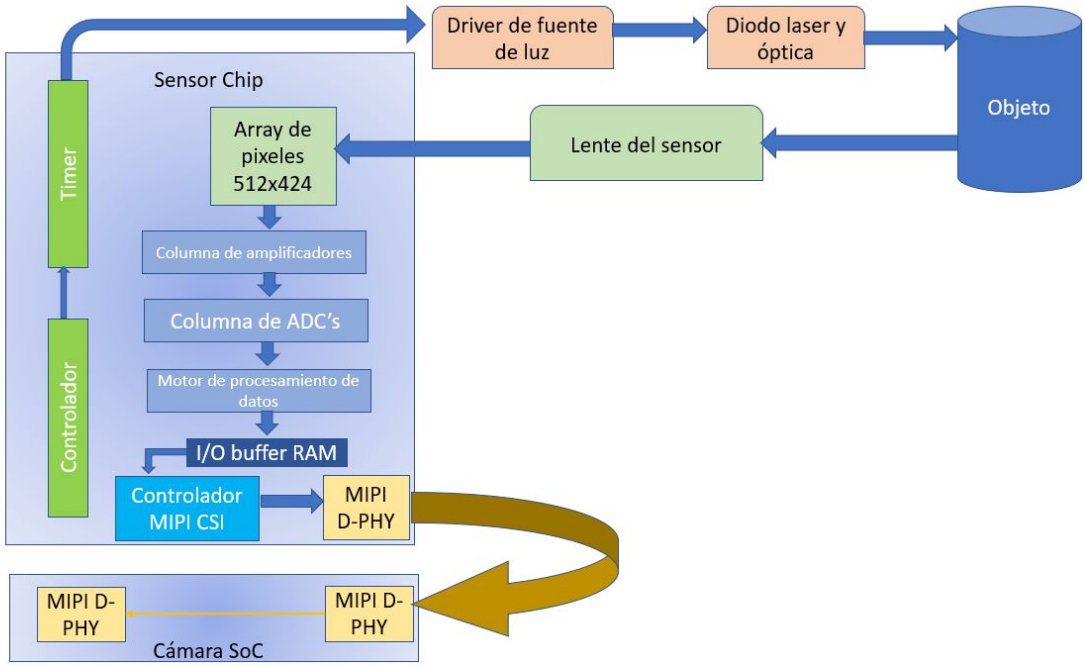


Figura 2.10: Arquitectura de la Kinect II [6]

Tal como se observa en la figura 2.11, el sensor Kinect II, incluye los siguientes elementos:

- Cámara RGB: se trata de una cámara la cual recoge información del espectro visible. Este tipo de información recogida no se emplea en este TFG para preservar la privacidad de los usuarios.

- Cámara IR: la cual recoge información en el espectro infrarrojo. Esta cámara permite obtener tanto la información de amplitud (imagen en escala de gris) como la de distancia.
- Iluminador IR: emisor que emite luz modulada en el espectro infrarroja. Esta luz no es visible por el ojo humano. Dicha luz es utilizada para poder medir la diferencia de fase entre el haz de luz emitido y el recibido y de esta manera se obtener la imagen de profundidad.

Figura 2.11: Hardware de la KinectII

2.3. Descriptores de características

Los descriptores de características permiten extraer información de una imagen. Hay una gran cantidad de descriptores que permiten extraer diferentes tipos de información de imágenes (RGB, profundidad, escala de gris, etc.). En este apartado se presentan algunos de los descriptores mas utilizados.

- **Descriptores de color**: una de las características de la visión humana es la detección de colores [14], por ello es importante el uso de este tipo de descriptores para poder distinguir algunos cambios de color que se puedan presentar.
- **Descriptores basados en los niveles de grises**: son usados cuando los cambios de contrastes son constantes. Se usan básicamente con imágenes las cuales solo tienen un canal de color (niveles de grises). Con estos descriptores se consigue normalizar regiones frente a traslaciones, rotaciones y cambios de escala.

Los descriptores también pueden estar basados en diferentes aspectos, como el perímetro, área, circularidad y rectangularidad. También hay mas aspectos en los que se pueden basar los descriptores, pero en este apartado se explicara los nombrados ya que son los mas representativos [15]:

- **Área y perímetro**: el descriptor correspondiente al área se basa en el numero de píxeles que hay en el interior del objeto. Mientras que el de perímetro se basa en los píxeles que rodean al objeto. Estos dos son descriptores muy simples, por lo que son solo usados en las imágenes binarias.

- **Circularidad y rectangularidad**: con este tipo de descriptores se consigue una información de la forma del objeto. Cuanto la proyección del objeto tenga una forma mas circular el descriptor de circularidad indicará dicha información dando un valor lo más alto posible. Con la rectangularidad se obtiene un valor elevado cuando el objeto tiene forma rectangular.

Existen descriptores más complejos [16] que permiten extraer información acerca de las características globales [17] o locales [18–20], así como otros específicos para las imágenes de profundidad [21–23] o RGBD [24, 25].

2.4. Clasificadores

Los clasificadores [26] son utilizados en el campo de la visión artificial como herramientas para poder etiquetar elementos que tengan diferencias entre si. Esto permite diferenciar los distintos elementos en diferentes clases, a partir de la información de entrada. Los clasificadores se pueden dividir en dos tipos principales en función del tipo de aprendizaje

- **Aprendizaje supervisado**: para llevar acabo este tipo de entrenamiento se utilizan datos etiquetados previamente , y con estos datos el usuario es el que debe indicar al algoritmo a que clase pertenecen. Algunos ejemplos son las máquinas de soporte vectorial (SVM) [27] o las redes neuronales [28] (figura 2.12), etc).

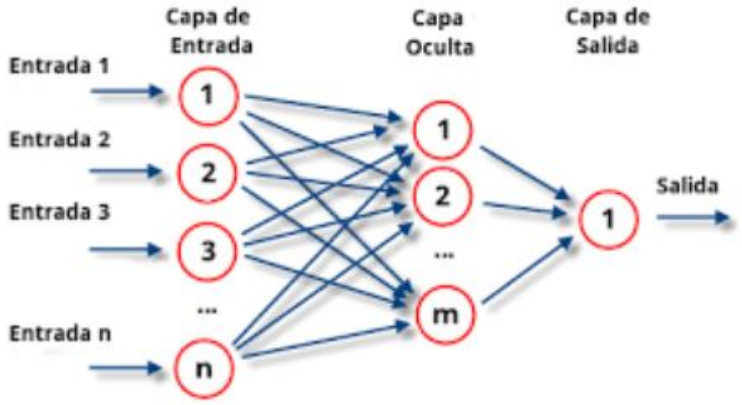


Figura 2.12: Ejemplo de sistema supervisado, una red neuronal

- **Aprendizaje no supervisado**: este método no dispone de datos etiquetados previamente, si no que se basa en sacar semejanzas entre los datos de entrada proporcionados, formando diferentes grupos. Algunos ejemplos son los métodos de *clustering* [29] como K-means.

En este TFG se utiliza un clasificador supervisado basado en el Análisis de Componentes Principales (PCA) [30] debido a que proporciona buenos resultados con un tiempo de procesamiento reducido. A continuación se describe el fundamento matemático de PCA con mayor detalle.

2.4.1. Análisis de componentes principales (PCA)

PCA (*Principal Components Analysis*) [10] está basado en la transformación lineal de los datos. Mediante esa transformación se genera un nuevo sistema de coordenadas. Estos sistemas de coordenadas se forman mediante la organización de estos datos en vectores de n elementos. Una vez organizados en estos vectores se procede a obtener la matriz de covarianza de los mismos, a partir de la que se extraen los autovectores y autovalores. Con dichos autovectores se puede empezar a comparar entre ellos, descartando aquellos que tienen un autovalor menor, eso indicara que son los menos significativos. Teniendo n dimensiones se obtendrían n autovectores y autovalores, y en este caso solo se seleccionarían m autovectores. Siendo $m \leq n$, de forma que es posible reducir la dimensión del conjunto de datos.

En primer lugar, se agrupan los datos obtenidos D, formando N vectores columna: $X_p^n = (X_n^1, X_n^2, \ldots, Xn^P)^T$ (donde n=1...,N), cada uno de ellos formados por P componentes (p = 1,...,P). A continuación, se calcula el vector media $\mu = (\mu_1, \mu_2, \ldots . \mu_P)^T$, definido por la siguiente ecuación:

$$\mu = (\frac{1}{N}) \sum_{n=1}^{N} X_n \tag{2.4}$$

Los N vectores, a los que se resta la media, se agrupan en la matriz R, en la que cada columna corresponde a un vector.

$$\mathbf{R} = \begin{pmatrix} \Phi_1 & \Phi_2 & \ldots & \Phi_N \end{pmatrix} \begin{pmatrix} X_1^1 - \mu_1 & X_2^1 - \mu_1 & ... & X_N^1 - \mu_1 \\ X_1^2 - \mu_2 & X_2^2 - \mu_2 & ... & X_N^2 - \mu_2 \\ ... & ... & ... & ... \\ X_1^P - \mu_P & X_2^P - \mu_P & ... & X_N^P - \mu_P \end{pmatrix} \tag{2.5}$$

A partir de las columnas de la matriz $\mathbf{R}$ (ecuación 2.5), se calcula la matriz de covarianza $\mathbf{C} \in \mathbb{R}.^{NxN}$ 2.6.

$$\mathbf{C} = \frac{1}{N} \sum_{j=1}^{N} (\Phi_j)(\Phi_j)^T \tag{2.6}$$

El siguiente paso es el cálculo de autovalores $(\lambda_1, \lambda_2, \ldots, \lambda_N)$ y autovectores $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N)$ de la matriz de covarianza $\mathbf{C}$, y ordenar los autovectores de mayor a menor valor de los autovalores asociados, para finalmente obtener la matriz de proyección $\mathbf{U} \in \mathbb{R}^{nxm}$, cuyas columnas son los autovectores asociados a los m mayores autovalores:

$$\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m) \tag{2.7}$$

PCA puede ser utilizado como clasificador [30], permitiendo determinar si un nuevo vector θ es similar, o no, a los vectores utilizados para crear la matriz de transformación para una determinada clase α: $\mathbf{U}_\alpha$. Para ello, el primer paso es proyectar el nuevo vector, al que previamente

se ha restado la media de la clase μ_α (ecuación 2.4), al espacio transformado PCA, utilizando para ello la expresión 2.8.

$$\widetilde{\theta}_\alpha = \mathbf{U}^T * (\theta - \mu_\alpha) \tag{2.8}$$

Posteriormente, se obtiene el vector recuperado $\widehat{\theta}_\alpha$ al que se le suma de nuevo la media, utilizando la ecuación 2.9.

$$\widehat{\theta}_\alpha = \mathbf{U} * \widetilde{\theta}_\alpha + \mu_\alpha \tag{2.9}$$

En caso de que el vector de entrada sea muy similar a los utilizados para obtener el modelo, el error de recuperación: definido como la diferencia entre el vector original θ y el recuperado $\widehat{\theta}_\alpha$, debe ser pequeño, mientras que si el vector es muy diferente, este error toma valores elevados.

Para el cálculo del error de recuperación existen diferentes alternativas, como la distancia Euclídea o la de Mahalanobis.

2.4.2. Distancia de Euclídea

Se denomina distancia euclídea al proceso matematico por el cual se puede encontrar la distancia entre dos puntos A(x_1, y_1) y B(x_2, y_2) en un espacio euclidiano, que parte del teorema de Pitágoras 2.10.

$$d = \sqrt[2]{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2.10}$$

La distancia entre un punto (x_1, y_1) y una recta d es la longitud del camino más corto que une el punto (x_1, y_1) con la recta d: Ax + By + C = 0, figura 2.13. Matemáticamente se expresa como 2.11:

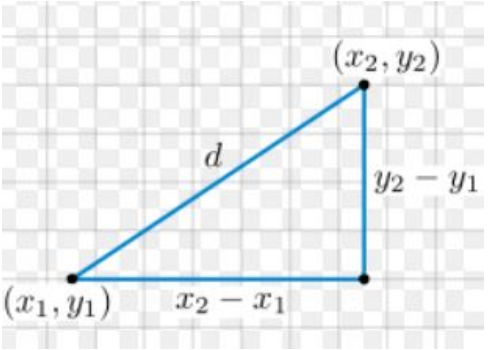


Figura 2.13: Ejemplo para calcular las distancias euclideas.

$$d = \frac{Ax_1 + By_1 + C}{\sqrt[2]{A^2 + B^2}} \tag{2.11}$$

2.4.2.1. Distancia de Mahalanobis

La Distancia de Mahalanobis fue descubierta por Mahalanobis en 1936 [31]. Lo práctico de esta medida es que determina la similitud entre dos variables aleatorioas con mas de una dimensión. Esta distancia tiene en cuenta la correlación de las variables aleatorias. Por ellos es mas sencillo trabajar con patrones los cuales no están separados de manera lineal, como se puede observar en el ejemplo de la figura 2.14.

La distancia de Mahalanobis [32] (d_{maha}) viene definida por la expresión 2.12. siendo $\overrightarrow{x}$ e $\overrightarrow{y}$ dos variables con la misma distribución de probabilidad y $\sum$ la matriz de covarianza.

$$d_{maha} = \sqrt{(\overrightarrow{x} - \overrightarrow{y})^T \sum (\overrightarrow{x} - \overrightarrow{y})} \tag{2.12}$$

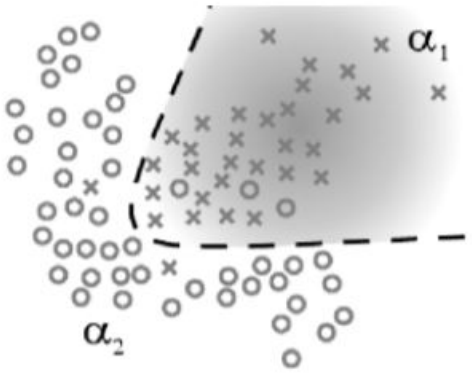


Figura 2.14: Ejemplo de representación de la región de separación de dos clases según el criterio de la distancia de Mahalanobis

Capítulo 3

Detección y clasificación de complementos

3.1. Introducción

En este capítulo se describe el proceso seguido para el desarrollo del software que permite cumplir los objetivos propuestos en este TFG: la detección y clasificación de complementos de una manera robusta.

En la figura 3.1 se presenta un diagrama general que muestra las principales etapas del algoritmo implementado. Esta figura ya se incluyó en el apartado 1.3 pero se repite aquí para facilitar la lectura del documento. A lo largo de este capítulo se describe en detalle cada una de las etapas mostradas en el diagrama.

Como ya se comentó en la introducción, para la realización de este TFG se ha partido del detector de personas desarrollado en los trabajos previos [9, 10]. Estos trabajos abordaban la detección robusta de personas a partir de la información de profundidad proporcionada por una cámara ToF en posición cenital. Tras la detección, en este trabajo se incorpora una etapa de clasificación de accesorios que vistan las personas. Dada la ubicación cenital del sensor, únicamente se clasifican los complementos que lleven en la cabeza: gorras, sombreros, etc.

Como se puede ver que en la figura 3.1, el sistema implementado consta de dos partes bien identificadas: una etapa *off-line*, en la que se lleva a cabo el entrenamiento del sistema, generando las clases para cada uno de los complementos a clasificar, y que sólo se ejecuta una vez. Y una etapa *on-line* en la que, por cada persona detectada, se realiza la clasificación de complementos.

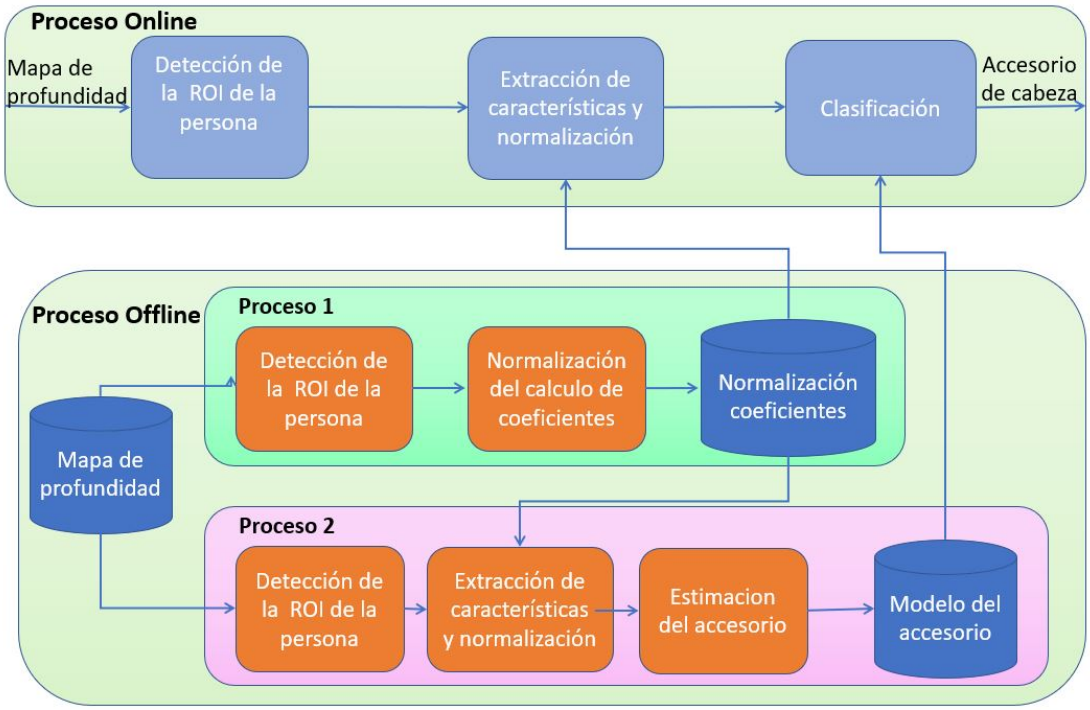


Figura 3.1: Diagrama de bloques general del sistema desarrollado en este TFG.

Cada una de las etapas del algoritmo se describe en detalle a lo largo de este capítulo. Cabe mencionar que hay etapas como el pre-procesado o la detección de personasque se ejecutan tanto en la parte *on-line* como *off-line*, por lo que solo se explica una vez

3.2. Detección de personas

En este apartado se describe la etapa de detección de personas, cuyo resultado es la ROI alrededor de cada persona detectada que posteriormente se utiliza como entrada para la clasificación de complementos. Esta etapa se describe en detalle en los trabajos previos [9, 10], por lo que aquí únicamente se incluye una breve explicación para la comprensión de este trabajo.

3.2.1. Pre-procesado de la imagen

Para la detección de personas, a partir de la información proporcionada por un sensor ToF en posición cenital, el primer paso es el pre-procesado de la imagen de profundidad. Como se ha explicado en el apartado 2.2.2, las cámaras ToF presentan diferentes fuentes de error, lo que provoca ruido y errores de medida en las imágenes adquiridas, tal como se puede observar en la imagen mostrada en la figura 3.2.

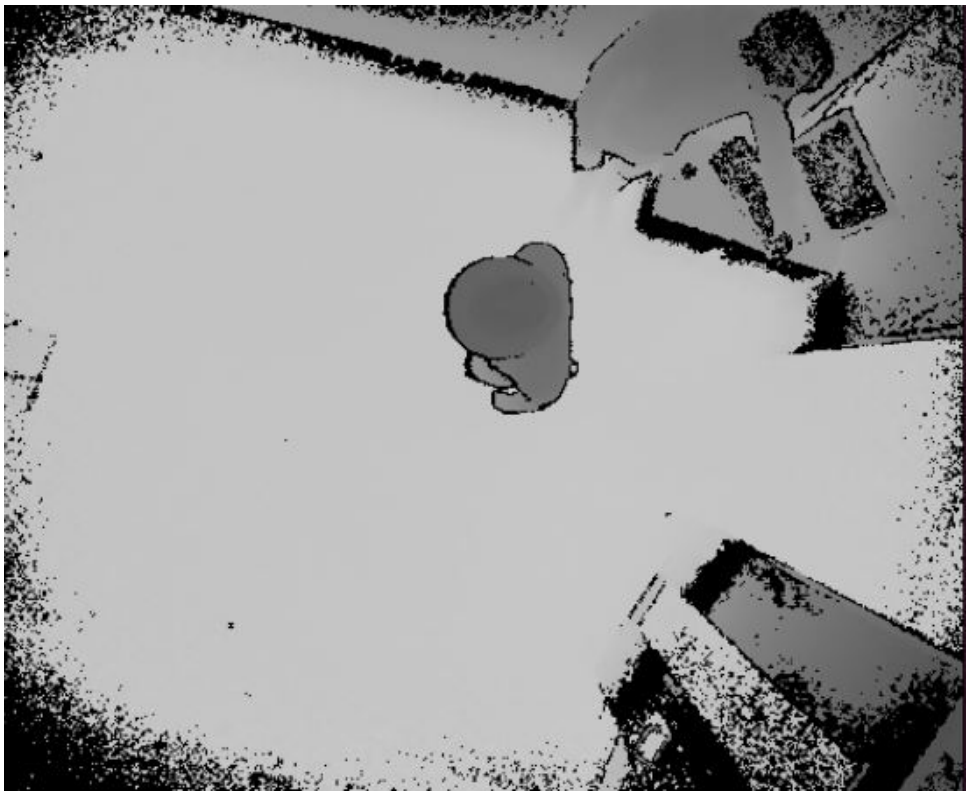


Figura 3.2: Ejemplo de imagen de entrada original (sin pre-procesado)

Como se puede ver en la figura 3.2, aparecen numerosos píxeles en color negro, esto se debe a que el sensor utilizado (Kinect II), asigna el valor 0 a aquellos píxeles en los que detecta una medida errónea. En esta figura se aprecian errores en los bordes de las personas y objetos (*flying pixels*), así como en las regiones más alejadas de la cámara (esquinas) debido a que la señal recibida tiene una amplitud insuficiente.

Para reducir el número de medidas erróneas, por cada píxel erróneo en la imagen se lleva a cabo una búsqueda de píxeles con valores válidos en su entorno de vecindad 2. En caso de que al menos uno de los píxeles vecinos tenga un valor válido, se le asigna al píxel erróneo el valor medio de todos los píxeles vecinos validos. El nivel de vecindad es 2 debido a que se considera que en ese entorno (2 píxeles en cada dirección), las medidas están fuertemente correladas entre sí.

Además, teniendo en cuenta que la cámara se encuentra en una posición cenital y elevada, y que la parte de la cual se desea sacar información es correspondientes los hombros y de la cabeza, se eliminan todas las medidas a una altura inferior a 0.5 metros desde el suelo. Con esto conseguimos una imagen con menos píxeles erróneos 3.3.

Figura 3.3: Ejemplo de imagen de entrada tras el pre-procesado.

3.2.2. Obtención de la ROI

Para la obtención de la ROI, y la detección de personas, se emplea el algoritmo implementado en [10]. En dicho algoritmo la imagen filtrada es dividida en subregiones todas del mismo tamaño. El tamaño de estas subregiones depende de la altura a la que se encuentra la cámara, la altura mínima de la persona y el área mínima que se quiera diferenciar.

Una vez dividida la imagen en subregiones se lleva acabo la localización y detección de los máximos deseados. Este algoritmo detecta los máximos de cada una de las subregiones, determinando cuáles de ellos corresponden a personas.

Como se ha comentado anteriormente la ROI debe contener información sobre la parte de la cabeza y de los hombros. Por este motivo, desde el máximo detectado (que es candidato a corresponder a la parte superior de la cabeza), únicamente se consideran aquellas medidas que se encuentran a una distancia inferior a $h_{interes}$ (ecuación 3.1), que en este trabajo se fija teniendo en cuenta consideraciones antropomórficas, para que la ROI incluya información de la cabeza y hombros [11].

$$h^{interest} = 40cm \tag{3.1}$$

Para obtener la ROI se establece un radio de vecindad asociado, L, de N niveles de vecindad.Para contar con todas las subregiones (SRs) que pertenecen al máximo localizado se evaluaran las 8 posibles direcciones. En la figura 3.4 se muestran los distintos niveles de vecindad, así como las 8 direcciones.

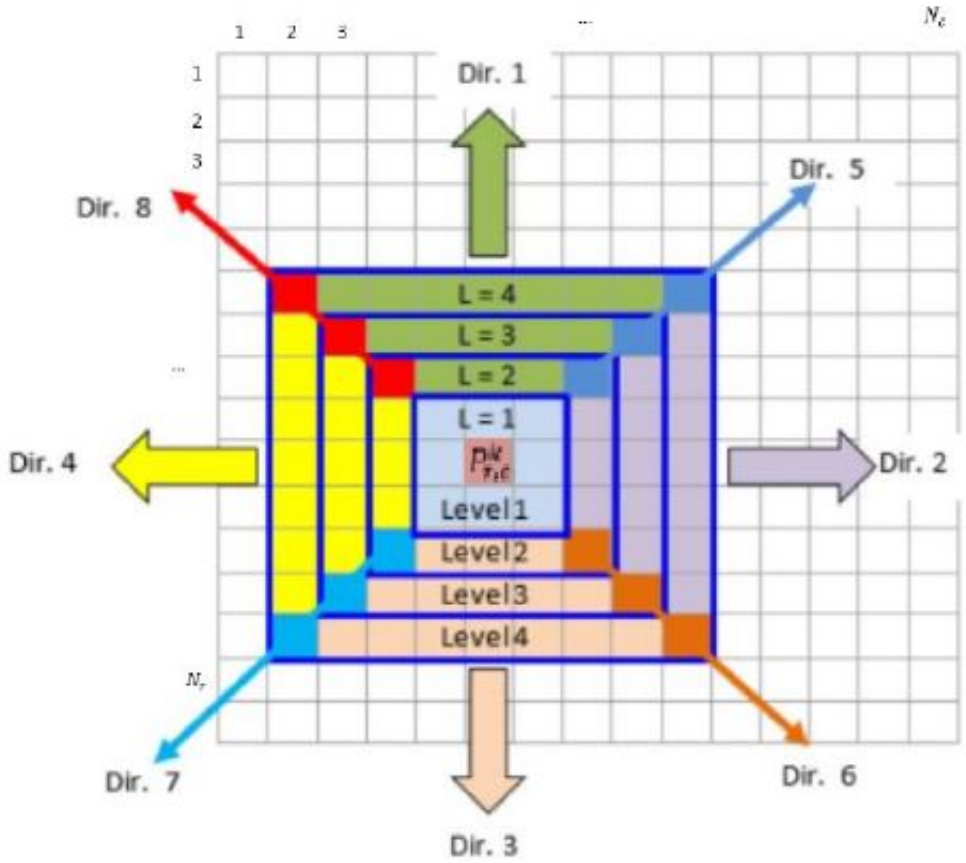


Figura 3.4: Niveles de vecindad y direcciones .

A continuación se describe el proceso de obtención de la ROI:

- El núcleo de la ROI, en la imagen anterior 3.4 corresponde con $SR^k_{r,c}$, es la zona en la que se ha localizado el máximo. $P^k_{r,c}$, donde r y c representan las coordenadas del pixel y k es el índice que indica el numero de máximo.

- El nivel de vecindad 1, L=1, esta compuesto por las 8 SR vecinas de $SR^k_{r,c}$. Si las SR del nivel 1 cumplen 3.2 pertenecen a la $ROI^k_{r,c}$

$$h^{max}_{SR} \geq h^{maxSR}_{r,c} - h^{interest} \tag{3.2}$$

- Dentro de estas direcciones se analizará las SR pertenecientes a los niveles L=2,3,4 ya que debido a la posición y características de la cámara el espacio que una persona no abarcará mas niveles. La pertenencia o no de una SR al máximo depende de cuatro premisas:

- Para que una SR perteneciente a un nivel L (L=2,. . . 4) en una dirección concreta 1, 2, 3 o 4 pertenezca a un máximo, el nivel inferior debe contener al menos L-1 SR en la misma dirección.

- Para que una SR perteneciente a un nivel L (L=2,. . . 4) en una dirección concreta 1, 2, 3 o 4 pertenezca a un máximo, la SR anterior en esa misma dirección contenga al máximo.

- El valor del máximo en las SR pertenecientes a un nivel L=2, 3 y 4 en una determinada dirección 1, 2, 3 o 4 , debe de cumplir la condición de la ecuación 3.2.

- Para diferenciar la ROI de dos individuos que se hayan muy próximos se ha de cumplir la siguiente ecuación 3.3.

$$h_{SR-1}^{max} \geq h_{SR}^{max} \geq hh_{SR+1}^{max} \tag{3.3}$$

- Las direcciones 5, 6, 7 y 8 pertenecen a las cuatro diagonales que parten desde $P_{r,c}^{k}$ como se puede ver en la figura 3.4 los niveles que se analizan son L=2, 3 y 4. Para que pertenezca a una SR se ha de cumplir las siguientes 3 restricciones:

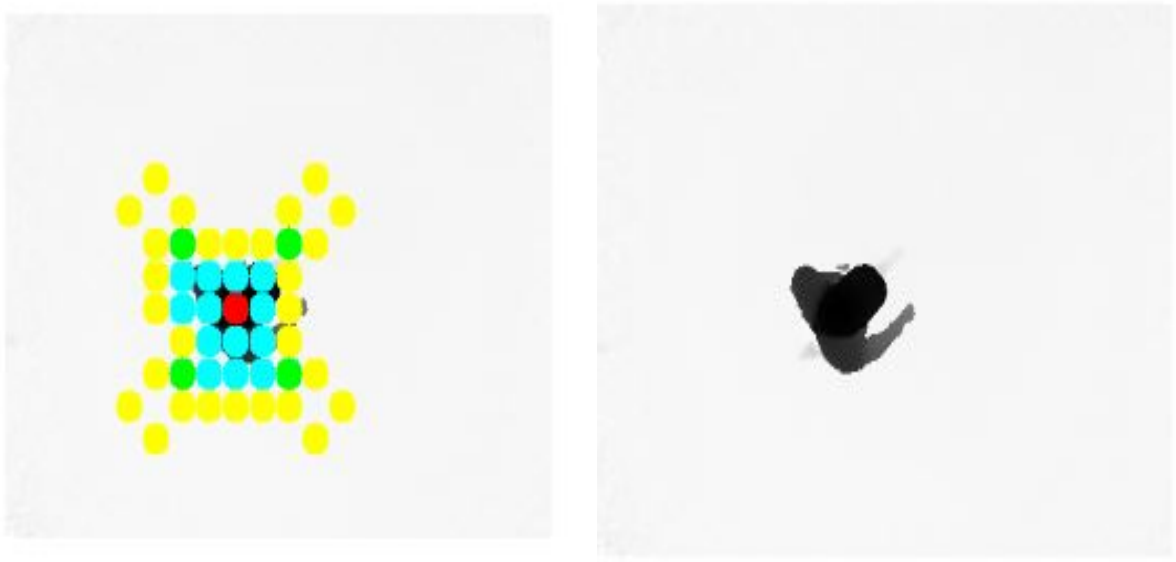
  - Para que una SR perteneciente a un nivel L (L=2,. . . 4) en una determinada dirección 1, 2, 3 o 4 pertenezca a un máximo,es necesario que la SR anterior en esa misma dirección pertenezca al máximo.

  - El valor del máximo en las SR pertenecientes a un nivel L=2, 3 y 4 en una determinada dirección 1, 2, 3 o 4 , debe de cumplir la condición de la ecuación 3.2.

  - Para diferenciar la ROI de dos individuos que se hayan muy próximos se ha de cumplir la siguiente ecuación 3.4 .

$$h_{SR-1}^{max} \geq h_{SR}^{max} \geq hh_{SR+1}^{max} \tag{3.4}$$

Para entender lo que se ha explicado de una manera mas visual, la figura 3.5a muestra las SRs analizadas para la imagen mostrada en la figura 3.5b. El punto rojo corresponde al centro el cual es el máximo localizado, en amarillo se muestran las direcciones 1, 2, 3 y 4 las cuales no cumplen los requisitos para pertenecer al máximo analizado. Los puntos verdes corresponden a las direcciones 5, 6, 7 y 8 donde tampoco se cumplen los requisitos necesarios para pertenecer a la ROI. Finalmente, los puntos azules son las SR correspondientes a las 8 direcciones expuestas que si cumplen los requisitos para pertenecer a la ROI del máximo analizado.



(a) Subregiones analizadas y ROI resultante. (b) Vista cenital persona.

Figura 3.5: Obtencion ROI mediante la búsqueda de SRs alrededor de un máximo.

3.3. Extracción de características

Para cada una de las ROIs definidas anteriormente se extrae un vector de características definido en función de la densidad de pixels asociadas a esta persona para diferentes valores de altura, dentro de la $h_{interes}$. Posteriormente, este vector se utiliza como entrada a la etapa de clasificación que permite diferenciar las personas de otros elementos en la escena. Como ya se ha comentado, esta etapa se desarrolló en el trabajo previo [10].

El vector de características desarrollado en el trabajo previo tiene seis componentes: los cinco primeros elementos del vector corresponden al número de puntos en distintas franjas correspondientes a la zona de la cabeza y hombros mientra que, la sexta se trata de la excentricidad de la franja superior. Utilizando este vector, es posible diferenciar entre personas y otros elementos de forma robusta, tal como se describe en [11]. Sin embargo, la información que contiene es insuficiente para la clasificación de complementos.

En el caso de este trabajo, se emplea un nuevo vector que, en vez de tener seis componentes tendrá ocho. Los cinco primeros elementos del vector incluyen información de la parte de la cabeza y las tres ultimas contienen información de la región de los hombros. De esta manera se genera un vector con mayor información para la clasificación de complementos. Para observar de una manera mas clara la interpretación del vector se emplea la siguiente figura 3.6. En ella se puede observar un ejemplo de las diferentes franjas a partir de las que se extrae el vector de características. En el eje de la izquierda se presenta la densidad de píxeles de profundidad cada dos centímetros en el eje de la izquierda. En el eje de la derecha se muestran los valores de las componentes del vector obtenidas a partir de la información de la izquierda. Comentar que esta imagen se trata de un ejemplo para clarificar las regiones, ya que lo que se observa es una imagen lateral de la persona, mientras que en este trabajo se emplea una vista cenital.

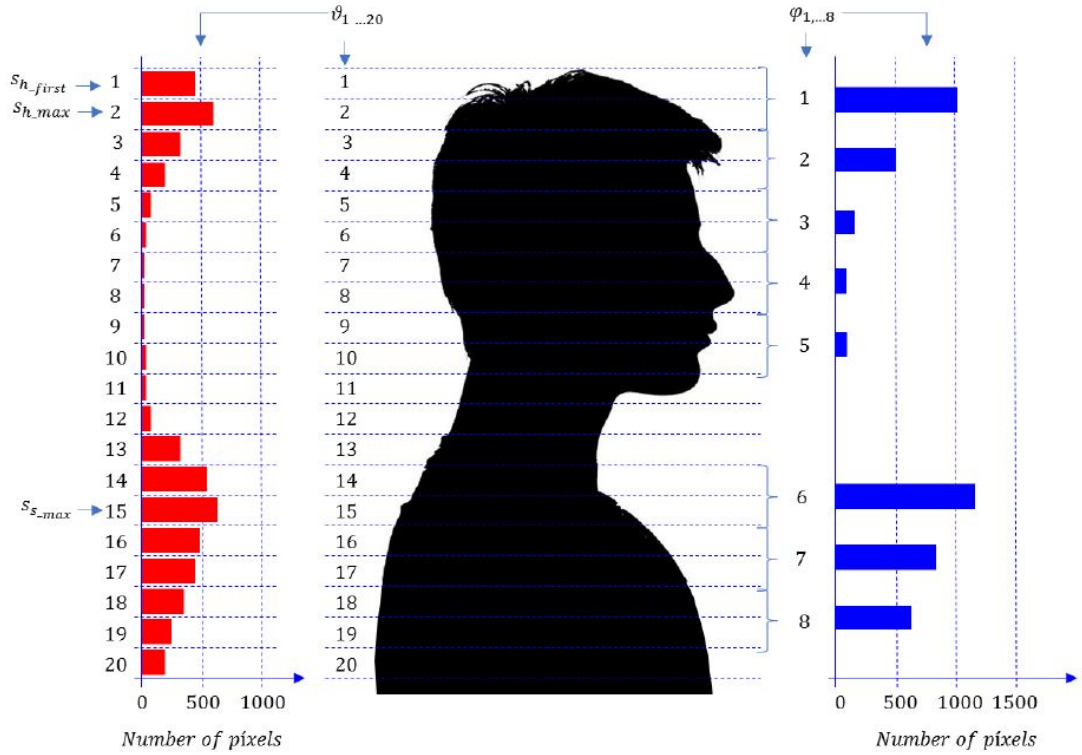


Figura 3.6: Densidad de los píxeles en las diferentes franjas de dos centímetros en que se divide la ROI [1]

Una vez explicado esto, a continuación se exponen los diferentes pasos para la generación y normalización de los vectores de características.

1. El primer paso es calcular los histogramas de profundidad. Como se puede observar en la imagen 3.6, se usan 20 intervalos de 2cm por intervalo a lo largo de la cabeza y hasta los hombros. Para cada una de esas franjas, se obtiene el número de medidas de profundidad en la ROI previamente obtenida. Esa información se almacena en un vector de 20 componentes θ=[θ_1,θ_2...θ_{20}]. θ_n cada una de las cuales corresponde al número de medidas en una de las franjas en las que se ha dividido la ROI.

2. A partir de la información anterior, se estima cuál de las franjas corresponde a la parte superior de la cabeza. Para ello se determina cuál de las tres primeras franjas tiene un mayor número de medidas: θ:$S_{h_max} = arg\max(\theta_1, \theta 2, \theta 3)$. 3.5.

$$f(x) = \begin{cases} \text{If } S_{hmax} = 1 : S_{hfirst} = S_{hmax} \\ \text{If } S_{hmax} > 1 and NT * S_{hmax} \geq S_{hmax} : S_{hfirst} = S_{hmax} - 1 \\ \text{If } S_{hmax} > 1 and NT * S_{hmax} \leq S_{hmax} : S_{hfirst} = S_{hmax} \end{cases} \tag{3.5}$$

El valor de NT ha sido obtenido de manera empírica, se trata de un valor entre 10 y 20. S_{hmax-1} debe ser NT veces mas grande que S_{hmax} para asumir que es el valor mas alto. Este paso se realiza para reducir el error en la estimación de la altura de la persona, ya que las cámaras ToF generan ruidos en los mapas de profundidad y debido que se desea

detectar accesorios y complementos pequeños necesitamos el menor ruido posible. De esta manera se puede calcular S_{hfirst} y con ello la altura.

$$h_p = h_{max} - 2cm * (S_{hfirst} - 1) \tag{3.6}$$

3. Las cinco primeras componentes del vector de características son las que describen la parte de la cabeza. Cada componente esta compuesta por la suma de dos elementos consecutivos de θ.

$$\varphi_i = \theta_{2i-2+S_{hfirst}} + \theta_{2i-1+S_{hfirst}}, fori = 1, 2, 3, 4, 5. \tag{3.7}$$

4. Las tres siguientes componentes del vector características en φ describen la estructura de los hombros. Cada componente esta compuesto de dos elementos consecutivos de θ organizados según la ecuación 3.8

$$\varphi_{i+5} = \theta_{2(i-1)-1+S_{Smax}} + \theta_{2(i-1)+S_{Smax}}, fori = 1, 2, 3. \tag{3.8}$$

5. Debido a que el número de píxeles tiene una gran dependencia de la altura de la persona, es necesario normalizar el vector de características (ψ) en función de dicha altura (ecuación 3.9) donde el valor entre el que se normaliza ($\widehat{\psi_1}$) se obtiene tal como se describe en el apartado 3.3.1.

$$\mu = \frac{\psi}{\widehat{\psi_1}} \tag{3.9}$$

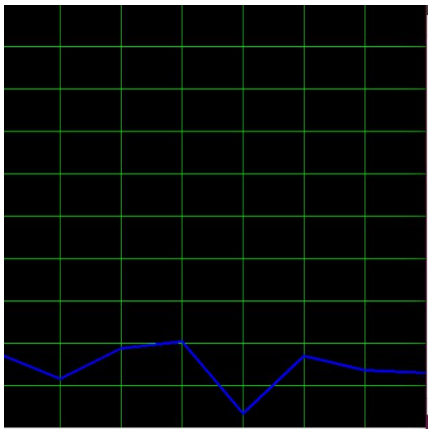
3.3.1. Calculo de los coeficientes de normalizacion

Como se ha comentado anteriormente, para que ψ sea independiente de la estatura de la persona (h_p), la relación entre h_p y ψ_1 debe ser calculada. Para generar dicho calculo se recogieron los vectores de características de un grupo de personas con distintas alturas, comprendidas entre 140cm y 213cm. A partir de los valores obtenidos de h_p y ψ_1 en los pasos anteriormente explicados, se llega a la siguiente relación cuadrática entre ellos.

$$\widehat{\psi_1} = a_0 + a_1 * h_p^2 + a_2 * h_p^2 \tag{3.10}$$

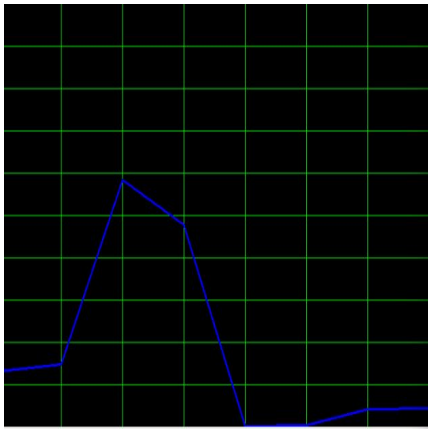
Donde a_0, a_1 y a_{20} son los coeficientes que deben de ser estimados. Usando el método de mínimos cuadrados no lineales se obtuvieron los siguientes valores: a_0 =1998.0, a_1 =-24.62 y a_2 =0.092 .

Una vez que las componentes del vector han sido normalizadas, éstas pueden representarse gráficamente. En la columna izquierda de figura 3.7 se muestran diferentes ejemplos del vector de características normalizado, para los mapas de profundidad mostrados en la columna derecha. En concreto, se presentan ejemplos para un sombrero pequeño (figura 3.7a), un sombrero grande (figura 3.7c) y una gorra (figura 3.7e)

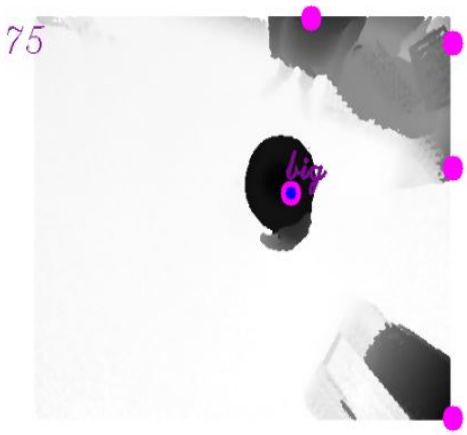


(a) Vector de características sombrero pequeño.

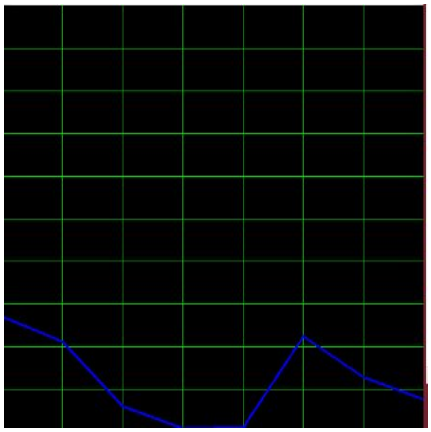
(b) Mapa de profundidad, sombrero pequeño.



(c) Vector de características sombrero grande.



(d) Mapa de profundidad, sombrero grande.



(e) Vector de características gorra.

(f) Mapa de profundidad, gorra.

Figura 3.7: Ejemplos de los vectores de características obtenidos para diferentes mapas de profundidad.

3.4. Clasificación de complementos

Para la clasificación de complementos se definen cinco clases, tal como se muestra en la figura 3.8: C_1: sin sombrero, C_2: gorra, C_3: sombrero pequeño, C_4: sombrero mediano y C_5: sombrero grande. Las diferentes clases se han organizado de forma jerárquica, pudiéndose realizar la clasificación con diferentes niveles de detalle, diferenciando entre dos clases: sin complementos/con complementos, tres clases: sin complementos, con gorra y con sombrero, y las cinco clases consideradas.

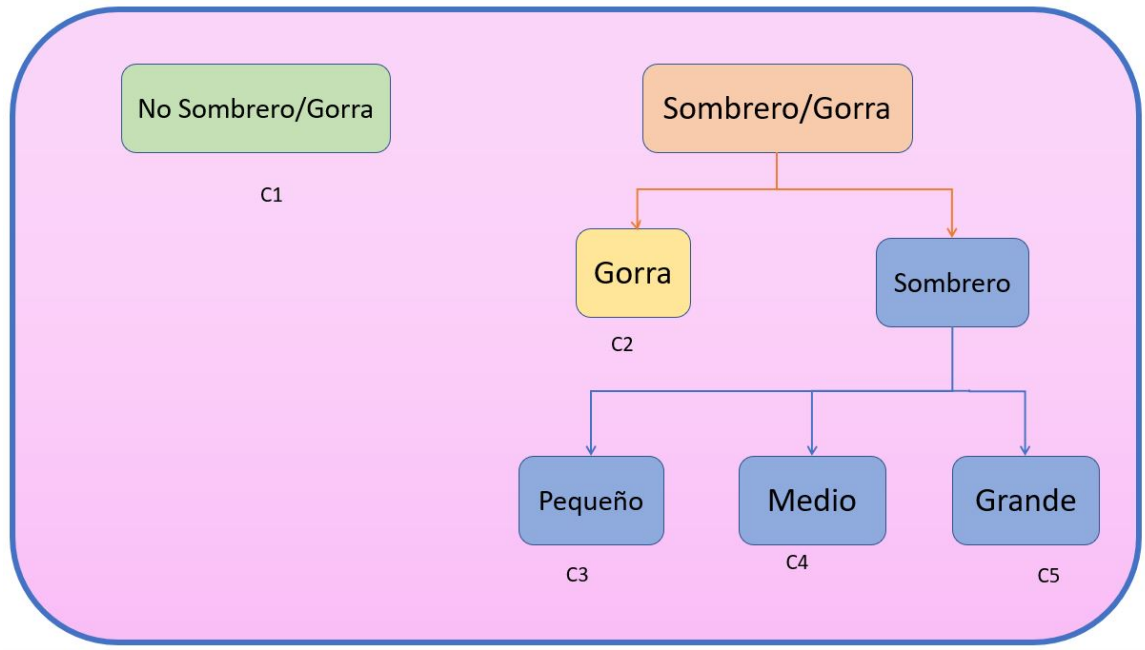


Figura 3.8: Diagrama de los distintos tipos de clases.

Para la clasificación se usa la técnica PCA por su simplicidad y baja carga computacional. PCA es conocido como un método que permite representar datos con un numero alto de dimensiones en espacios de bajas dimensiones y por la facilidad que ofrece a la hora de la clasificación.

El clasificador PCA requiere un entrenamiento para estimar un modelo para cada una de las clases definidas previamente: C_n donde $n = 1, 2, \ldots, 5$. Cada modelo está compuesto de un vector media $\widehat{\mu_n}$, una matriz de covarianza $\mathbf{C_n}$ y una matriz de transformación $\mathbf{U_n}$, tal como se ha explicado en el apartado 2.4.1. Para la obtención de la matriz de transformación $\mathbf{U}$ es necesario determinar el número de componentes del espacio transformado PCA (m). En este caso, se ha comprobado que las tres primeras componentes contienen la información principal de cada clase, por lo que la matriz de transformación se obtiene a partir de los tres autovectores asociados a los mayores autovalores.

Como se ha comentado en el apartado 2.4.1, para determinar si un nuevo vector de características θ pertenece a una determinada clase α, es necesario calcular la diferencia entre el vector original θ y el vector recuperado para la clase bajo estudio $\widehat{\theta_\alpha}$ (obtenido tras proyectar dicho vector al espacio transformado PCA mediante la ecuación 2.8 y recuperarlo posteriormente al espacio original utilizando la expresión 2.9).

Para comparar estos vectores existen diferentes alternativas, siendo las más utilizadas la distancia euclídea y la de distancia de Mahalanobis. En este trabajo se han considerado ambas posibilidades, presentando resultados tanto para el caso de utiliza la distancia euclídea, como para la distancia de Mahalanobis.

Durante la ejecución del software se lleva acabo esta diferencia entre el vector recuperado y el nuevo para cada clase. De esta manera se entregan cinco resultados, cada resultado obtenido muestra el error de recuperación, como se ha explicado en el apartado anterior 2.4.1 cuanto mas pequeño sea este error mas similar será el nuevo vector a la clase correspondiente. Por ello, se asigna el nuevo vector θ a la clase cuyo error de recuperación (diferencia entre el vector de entrada θ y el vector recuperado $\widehat{\theta}_\alpha$) sea menor.

Capítulo 4

Resultados experimentales

En este capítulo se presentan los resultados obtenidos durante la realización de este TFG. Para ello, en primer lugar se describen las secuencias de imágenes de profundidad y los distintos complementos utilizados para llevar a cabo la validación experimental del software desarrollado. Posteriormente se presentan las métricas utilizadas para la evaluación de los resultados, para finalizar mostrando y justificando los principales resultados experimentales.

4.1. Escenario de evaluación experimental

Para la evaluación del sistema desarrollado en este TFG se ha utilizado un conjunto de secuencias [33] grabadas utilizando dos sensores Kinect II y etiquetas por los miembros del grupo de investigación GEINTRA [8], cuyas características se detallan más adelante.

Los diferentes tipos de complementos que se han considerado en la clasificación, así como las clases a las que pertenecen, se muestran en la figura 4.1. Por cada uno de ellos se incluyen las dimensiones en centímetros. Además, en esta tabla, por cada una de las clases se indica qué complementos (gorra o sombrero) se han utilizado para la etapa de entrenamientos (en la primera columna), y cuáles para la validación.

Los diferentes complementos considerados se dividen en 5 clases diferentes,

- Clase C_1: sin complementos.
- Clase C_2: con gorra.
- Clase C_3: con sombrero pequeño.
- Clase C_4: con sombrero mediano.
- Clase C_5: con sombrero grande.

Figura 4.1: Diferentes tipos de complementos considerados en este TFG. [1]

Las clases anteriores se agrupan de forma jerárquica tal como se muestra en la figura 4.2. Esta figura ya se incluyó en el apartado 3.4 pero se repite aquí para facilitar la lectura del documento. De acuerdo a esta clasificación, se presentan resultados para dos (sin complementos/con complementos), tres (sin complementos/con gorra/con sombrero) y cinco clases.

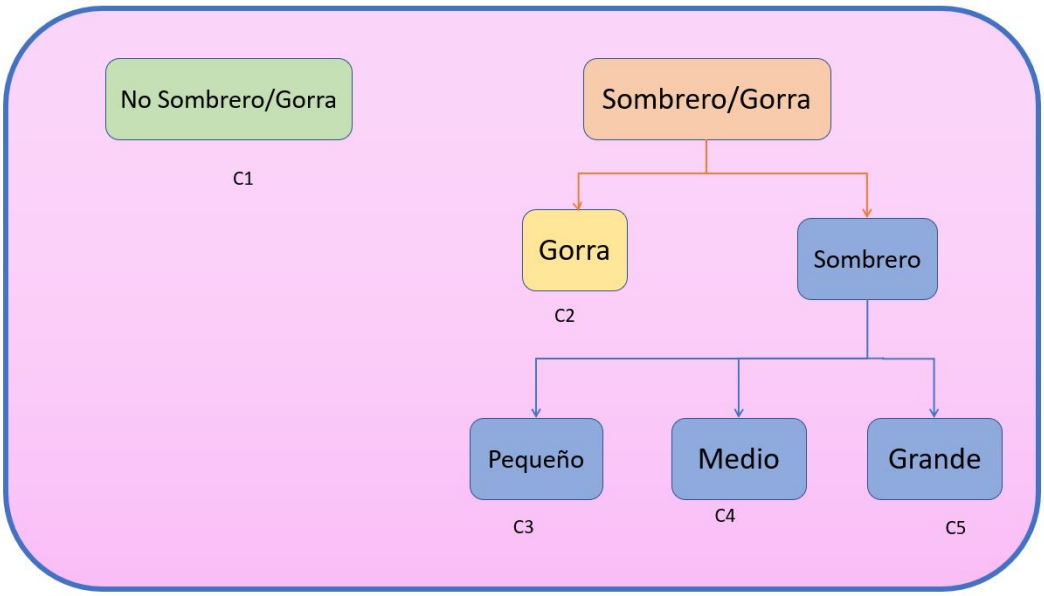


Figura 4.2: Diagrama de las distintas clases de complementos consideradas en este TFG.

El conjunto de secuencias utilizado para este TFG tiene las siguientes características:

- Las secuencias utilizadas incluyen una sola persona en la escena ya que el objetivo no es la detección de personas, sino la clasificación de complementos.
- Se incluyen grabaciones con diferentes personas que tienen distintas alturas. En concreto, se han realizado grabaciones con 30 personas diferentes cuyas alturas varían entre 1.6 y 2.1 m. Las secuencias correspondientes a las diferentes personas grabadas se han dividido en dos conjuntos, uno para la etapa de entrenamiento, y otro para la validación.
- Las personas visten algunos de los sombreros mostrados en la tabla 4.1, o aparecen sin complementos y con diferentes peinados: pelo largo, pelo corto, recogido en una coleta, etc.
- Los datos disponibles se han dividido de forma que los complementos que se usan en las secuencias de entrenamiento, no son utilizados para la evaluación del sistema.
- Las grabaciones han sido realizadas con dos cámaras en posición cenital ubicadas en el espacio inteligente (ISPACE) del grupo GEINTRA [8] de la Universidad de Alcalá, en las posiciones indicadas en el esquema de la figura 4.3.Ambas cámaras se encuentran a una altura de 3.40m del suelo, fijadas al techo tal como se muestra en la figura 4.4.

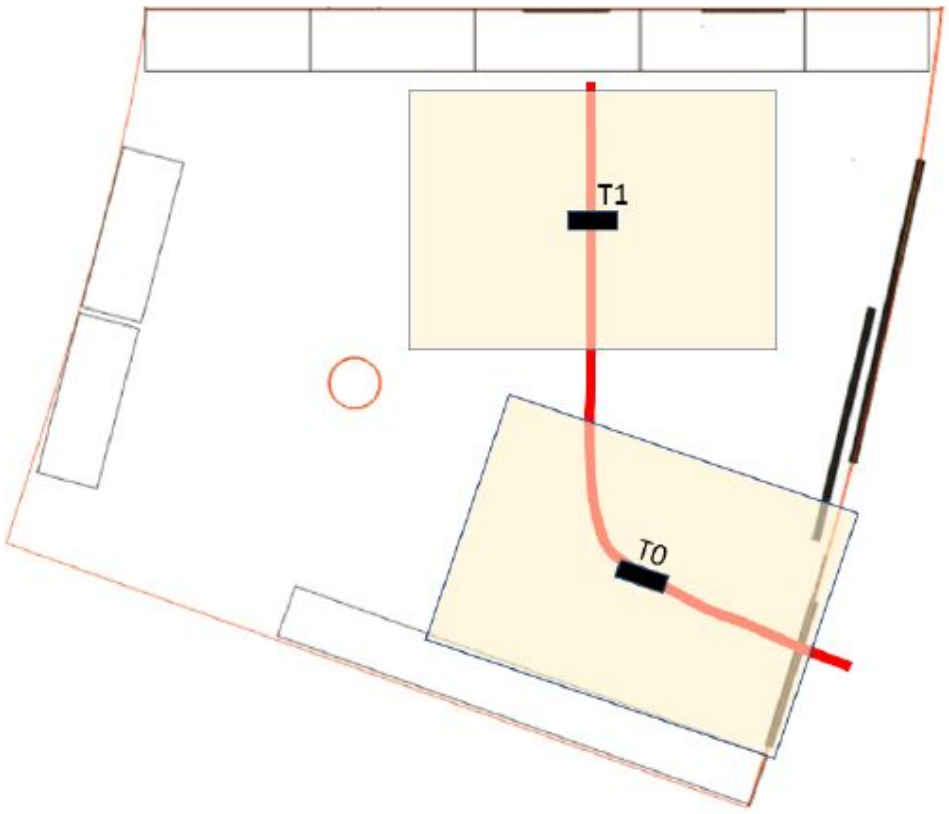


Figura 4.3: Posición de las cámaras en el laboratorio.

Figura 4.4: Cámara usada para detección de las imágenes.

Para la etapa de entrenamiento del sistema se han empleado imágenes adquiridas tanto con la cámara T0 como con T1. La tabla 4.1 muestra el número de personas e imágenes utilizadas para el entrenamiento de cada una de las clases.

	Sin Gorra/ Sombrero	Gorra	Sombrero pequeño	Sombrero mediano	Sombrero Grande
Nº sujetos	20	3	3	5	3
Nº imágenes	2470	1128	1094	1295	1144

Tabla 4.1: Número de personas diferentes e imágenes utilizadas para el entrenamiento de cada una de las clases.

En el proceso de evaluación se cuenta con secuencias de imágenes de los diferentes usuarios vistiendo los diferentes accesorios. Cabe destacar que los complementos incluidos en la etapa de validación no se han utilizado para el entrenamiento, tal como se observa en la figura 4.1. En la tabla 4.2 se muestra el numero de imágenes y usuarios asociados a cada clase en el proceso de validación. En dicha tabla se hace una diferenciación entre las cámaras T0 y T1, ya que cada una de ellas se ha evaluado de forma independiente con el objetivo de determinar la influencia de la posición de la cámara en la clasificación.

		Sin Gorra/ Sombrero	Gorra	Sombrero pequeño	Sombrero mediano	Sombrero grande
T0	Nº sujetos	17	17	17	17	17
	Nº imágenes	4737	7630	6750	13527	6726
T1	Nº sujetos	16	17	17	17	17
	Nº imágenes	3688	7264	6556	13811	6693

Tabla 4.2: Número de usuarios e imágenes utilizadas en la etapa de validación para cada una de las clases.

4.2. Métricas empleadas para la evaluación del sistema

Se han empleado matrices de confusión para poder estudiar el ratio de aciertos del sistema. Este tipo de matrices son empleadas comúnmente en el mundo de la visión artificial por la gran cantidad de información que proporciona de una manera rápida y sencilla. En el caso de un clasificador binario (con dos clases) la matriz de confusión tiene cuatro elementos, tal como se muestra en la figura 4.5.

Las matrices de confusión se pueden dividir en cuatro elementos característicos:

- Verdaderos positivos: es la cantidad de positivos que fueron clasificados correctamente como positivos.
- Falsos positivos: es la cantidad de negativos que fueron clasificados incorrectamente como positivos.
- Verdaderos negativos: es la cantidad de negativos que fueron clasificados correctamente como negativos.
- Falsos negativos: es la cantidad de positivos que fueron clasificados incorrectamente como negativos.

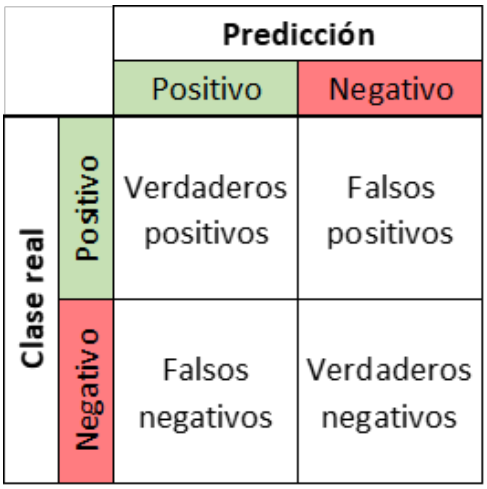


Figura 4.5: Ejemplo de matriz de confusión para un clasificador binario.

Pero en el caso de que se este estudiando la clasificación con mas de dos clases, la matriz de confusión es una matriz cuadrada de nxn elementos (siendo n el número de clases) que aporta información de la tasa de aciertos y la confusión entre clases.

En todas las matrices de confusión se muestran los porcentajes de detección para cada una de las clases reales. Además, para los elementos de la diagonal principal se incluye también el intervalo de confianza al 95 % definido según la ecuación 4.1, donde p indica el porcentaje de aciertos y ns el número de muestras consideradas.

$$1{,}96 * \sqrt{\frac{p * (100 - p)}{ns}} \tag{4.1}$$

4.3. Resultados

En el caso de este TFG, a continuación se muestran los resultados para los diferentes niveles de detalle mostrados en la figura 4.2, incluyendo las siguientes situaciones:

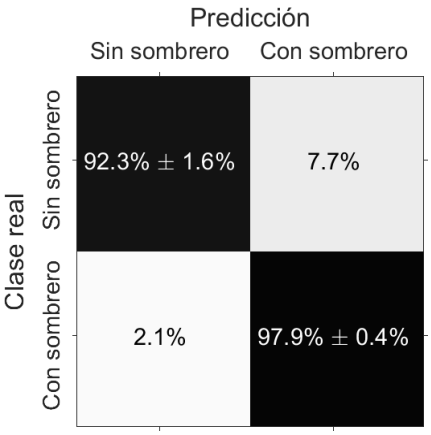
- **Dos clases**: sin complementos (C_1) y con complementos (que agrupa las clases C_2, C_3, C_4 y C_5).
- **Tres clases**: sin complementos (C_1), con gorra (C_2) y con sombrero (C_3, C_4 y C_5)
- **Cuatro clases**: sin complementos (C_1), con gorra (C_2), con sombrero pequeño (C_3 y C_4) y con sombrero grande (C_5)
- **Cinco clases**: que incluye todas las clases consideradas en el TFG de forma independiente.

Para determinar las diferencias debidas a la posición y orientación de la cámara se muestran por separado los resultados correspondientes a secuencias grabadas con la cámara T_0 y las grabadas con T_1. Además, se comparan los resultados empleando las dos alternativas consideradas para el cálculo del error de recuperación: distancia euclídea y de Mahalanobis.

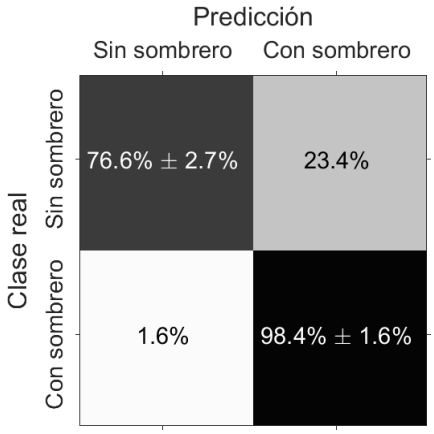
4.3.1. Resultados con dos clases

En este apartado se comprueba la matriz de confusión para el caso de un clasificador binario, que determinar si la persona lleva, o no, algún complemento.

La figura 4.6 muestra las matrices de confusión obtenidas utilizando la distancia euclídea (figura 4.6a) y de Mahalanobis (figura 4.6b) para las secuencias de validación adquiridas con la cámara T0, mientras que la figura 4.7 presenta los resultados para las secuencias de la cámara T1.

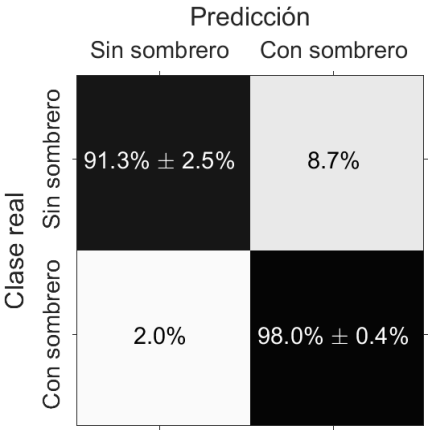


(a) Matriz de confusión obtenida utilizando la distancia Euclídea.

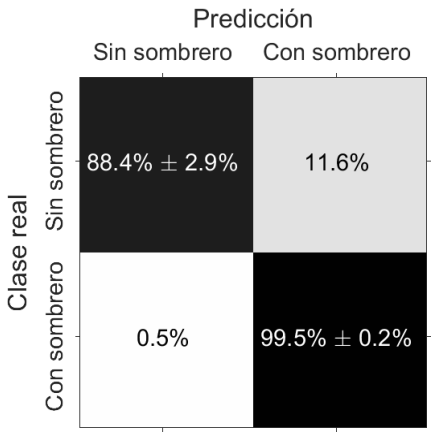


(b) Matriz de confusión obtenida utilizando la distancia de Mahalanobis.

Figura 4.6: Resultados obtenidos para las secuencias de validación adquiridas con la cámara T0 pra la clasificación con dos clases.



(a) Matriz de confusión obtenida utilizando la distancia Euclídea.



(b) Matriz de confusión obtenida utilizando la distancia de Mahalanobis.

Figura 4.7: Resultados obtenidos para las secuencias de validación adquiridas con la cámara T1 para la clasificación con dos clases.

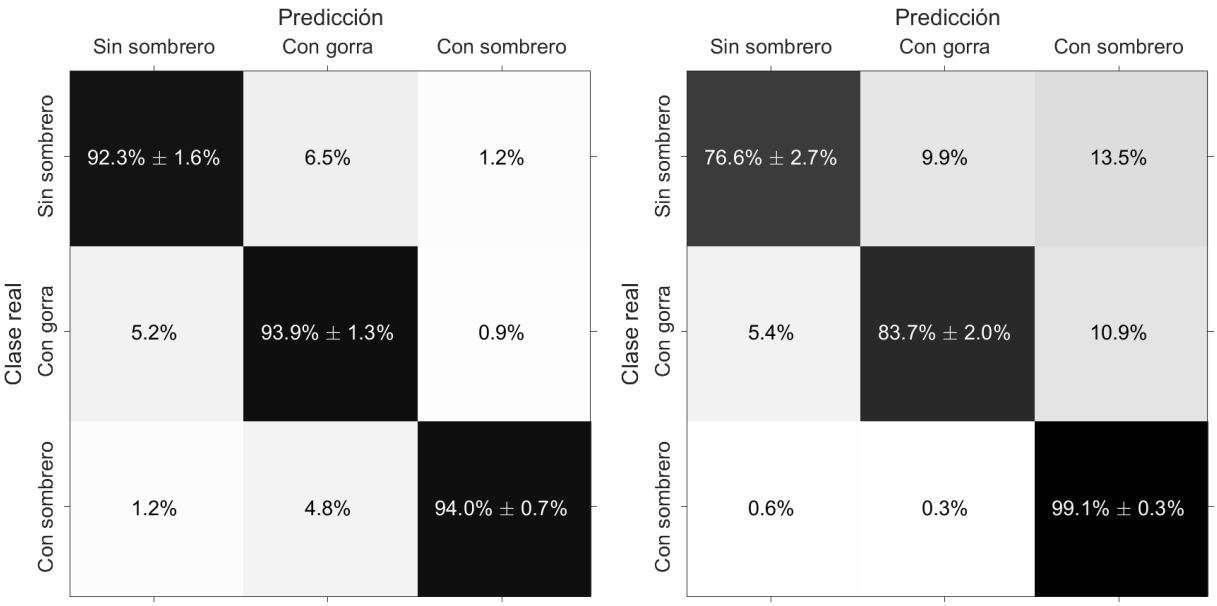
Los resultados mostrados permiten concluir que se obtienen mejores resultados utilizando la distancia euclídea. También se comprueba que el sistema es independientemente del ángulo de grabación, siendo los resultados similares para ambas cámaras. En la tabla 4.3 se muestran los resultados obtenidos utilizando la distancia euclídea para cada una de las cámaras, así como los resultados totales.

		Predicción		
Cámara	Clase real	Sin gorra/sombrero	Con gorra/sombrero	Total
T0	Sin gorra/sombrero	928	77	1005
	Con gorra/sombrero	128	5941	6069
T1	Sin gorra/sombrero	455	43	498
	Con gorra/sombrero	80	3841	3921
Total	Sin gorra/sombrero	1383	120	1503
	Con gorra/sombrero	208	9782	9990

Tabla 4.3: Resultados obtenidos para la clasificación con dos clases utilizando la distancia euclídea.

4.3.2. Resultados con tres clases

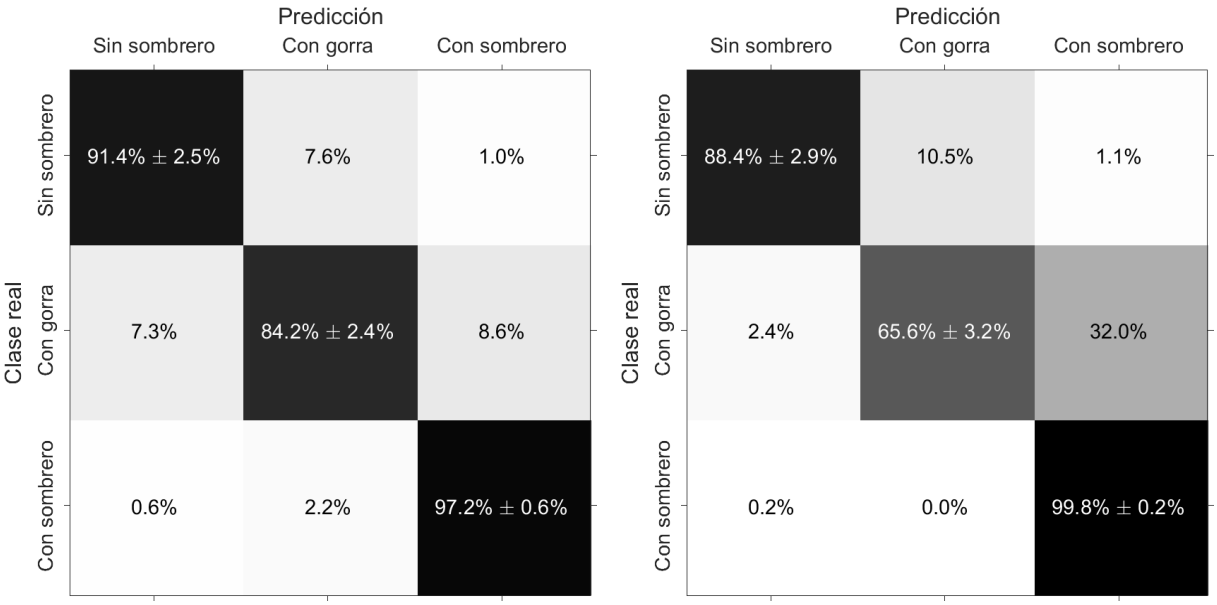
En este apartado se estudia el funcionamiento del sistema cuando clasifica tres clases distintas: sin complemento, con gorra y con sombrero. Al igual que en el caso de dos clases, se presentan tanto los resultados para la cámara T0 (figura 4.8) como para T1 (figura 4.9), así como para ambas alternativas de cálculo del error de recuperación.



(a) Matriz de confusión obtenida utilizando la distancia Euclídea para el cálculo del error de recuperación.

(b) Matriz de confusión obtenida utilizando la distancia de Mahalanobis para el cálculo del error de recuperación.

Figura 4.8: Resultados obtenidos para las secuencias de validación adquiridas con la cámara T0 para la clasificación en tres clases de complementos.



(a) Matriz de confusión obtenida utilizando la distancia Euclídea para el cálculo del error de recuperación.

(b) Matriz de confusión obtenida utilizando la distancia de Mahalanobis para el cálculo del error de recuperación.

Figura 4.9: Resultados obtenidos para las secuencias de validación adquiridas con la cámara T1 para la clasificación en tres clases de complementos.

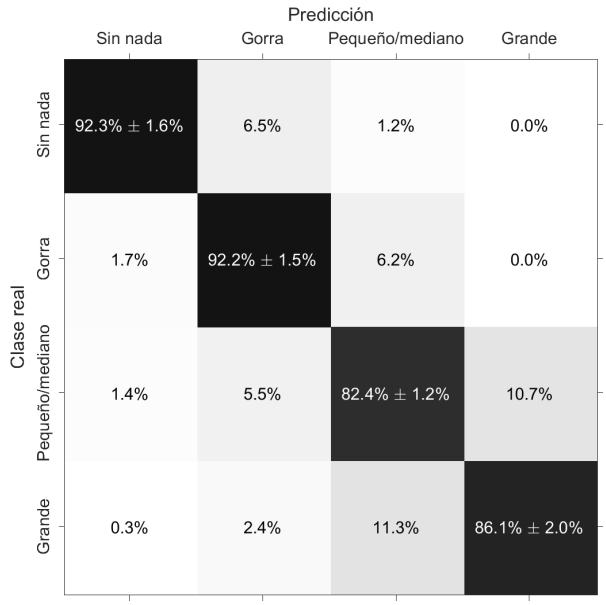
En la tabla 4.4 se muestran los resultados obtenidos para el caso de tres clases utlilizando la distancia euclídea para el cálculo del error de recuperación.

		Predicción			
Cámara	Clase real	Sin gorra/sombrero	Con gorra	Con sombrero	Total
T0	Sin gorra/sombrero	928	65	12	1005
	Con gorra	71	1282	12	1365
	Con sombrero	55	225	4424	4704
T1	Sin gorra/sombrero	455	38	5	498
	Con gorra	63	728	74	865
	Con sombrero	17	68	2971	3056
Total	Sin gorra/sombrero	1383	103	17	1503
	Con gorra	134	2010	86	2230
	Con sombrero	72	293	7395	7760

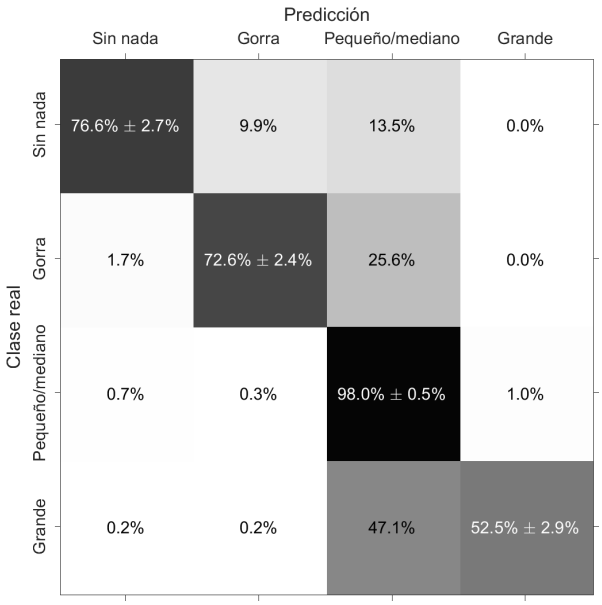
Tabla 4.4: Resultados obtenidos para la clasificación con dos clases utilizando la distancia euclídea.

4.3.3. Resultados con cuatro clases

En este apartado se estudia el funcionamiento del sistema cuando clasifica cuatro clases distintas. En la figura 4.10 se muestran los porcentajes de acierto con T0 utilizando tanto la distancia de euclidea como la de Mahalanobis. En dicha clasificación las clases C3 y C4 se han unificado ya que la diferencia entre un sombrero pequeño y uno grande es difícil de apreciar.



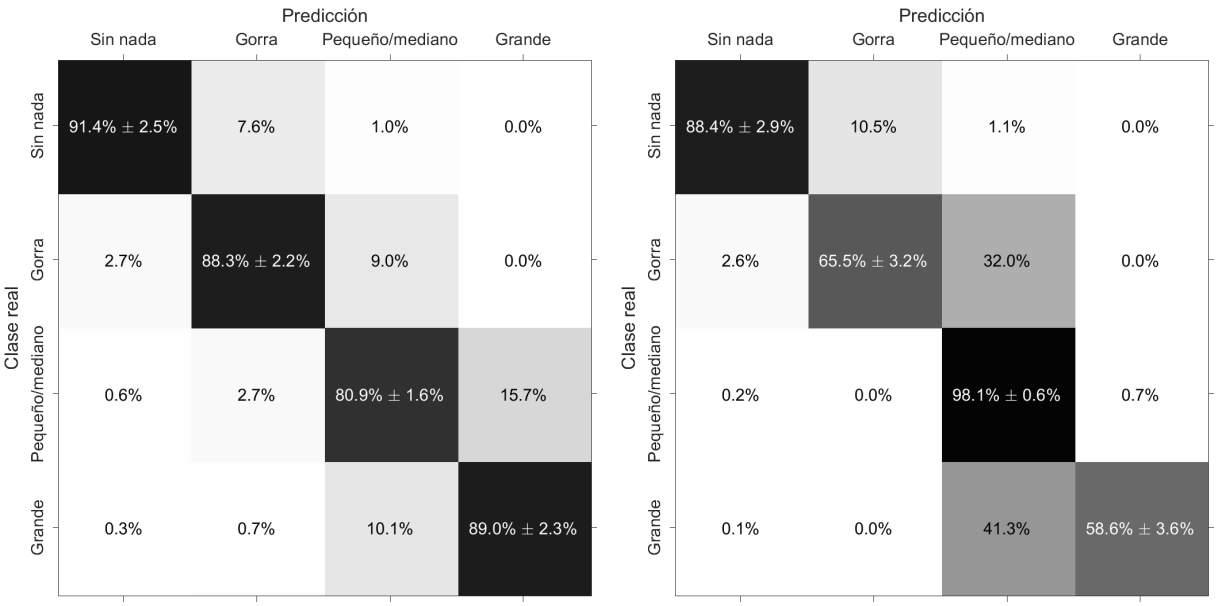
(a) Matriz de confusión obtenida utilizando la distancia Euclídea para el cálculo del error de recuperación.



(b) Matriz de confusión obtenida utilizando la distancia de Mahalanobis para el cálculo del error de recuperación.

Figura 4.10: Resultados obtenidos para las secuencias de validación adquiridas con la cámara T0 para la clasificación en cuatro clases de complementos.

En la imagen 4.11 se muestran los resultados para la cámara T1.



(a) Matriz de confusión obtenida utilizando la distancia Euclídea para el cálculo del error de recuperación.

(b) Matriz de confusión obtenida utilizando la distancia de Mahalanobis para el cálculo del error de recuperación.

Figura 4.11: Resultados obtenidos para las secuencias de validación adquiridas con la cámara T1 para la clasificación en cuatro clases de complementos.

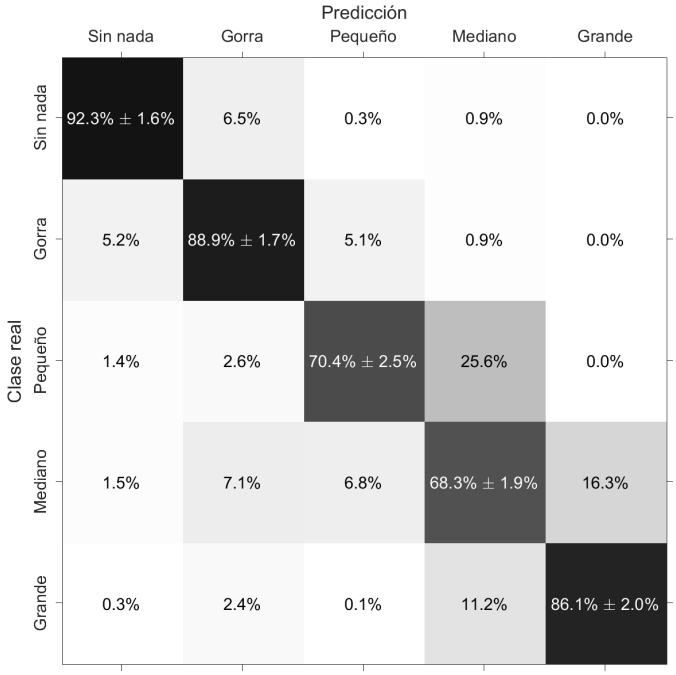
La tabla 4.5 recoge los resultados para cuatro clases. Se observa que el numero imágenes clasificadas de manera correcta ha disminuido, esto se debe a que se ha aumentado la dificultad a la hora de clasificar al añadir una clase mas.

		Predicción				
Cámara	Clase real	Sin compl.	Gorra	S. peq/med	S. grande	Total
T0	Sin compl.	928	65	12	0	1005
	Gorra	22	1213	81	0	1365
	S. peq/med	52	199	2969	384	3604
	S. grande	3	26	124	947	1100
T1	Sin compl.	455	38	5	0	498
	Gorra	63	728	74	0	865
	S. peq/med	15	63	1879	365	2322
	S. grande	2	5	74	653	734
Total	Sin compl.	1383	103	17	0	1503
	Gorra	85	1941	155	0	2181
	S. peq/med	67	262	4848	749	5926
	S. grande	5	31	198	1600	1834

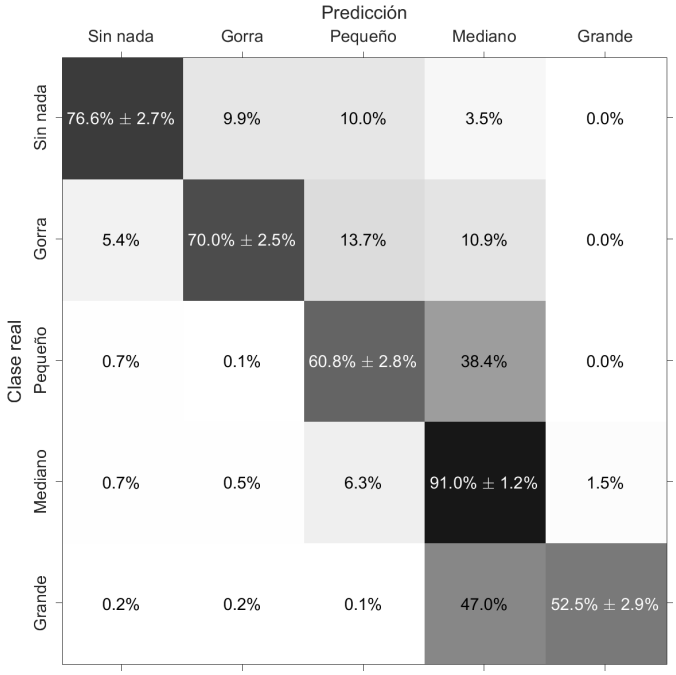
Tabla 4.5: Resultados obtenidos para la clasificación con tres clases utilizando la distancia euclídea.

4.3.4. Resultados con cinco clases

Finalmente, en este apartado se presentan los resultados para el caso más complejo, en el que se contemplan las cinco posibles clases. La figura 4.12 muestra los resultados para la cámara T0, mientras que en la figura 4.13 se presentan los resultados para T1.

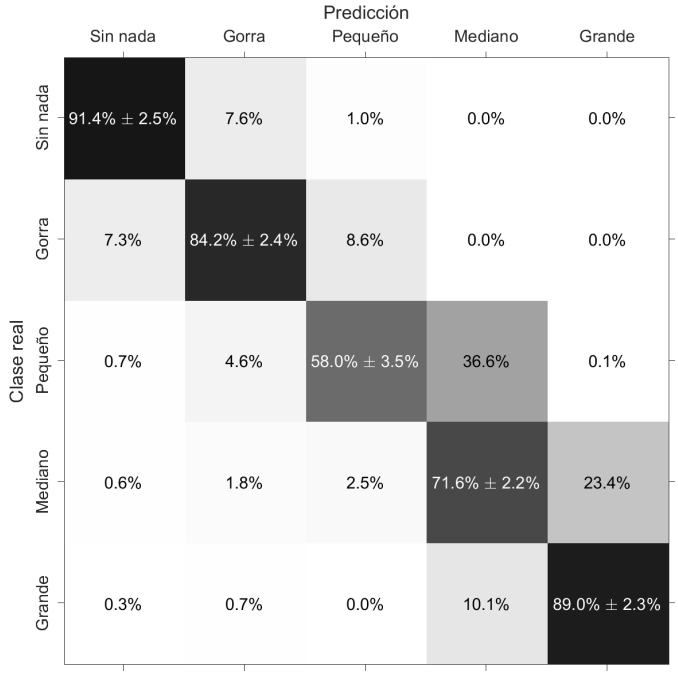


(a) Matriz de confusión obtenida utilizando la distancia Euclídea para el cálculo del error de recuperación.

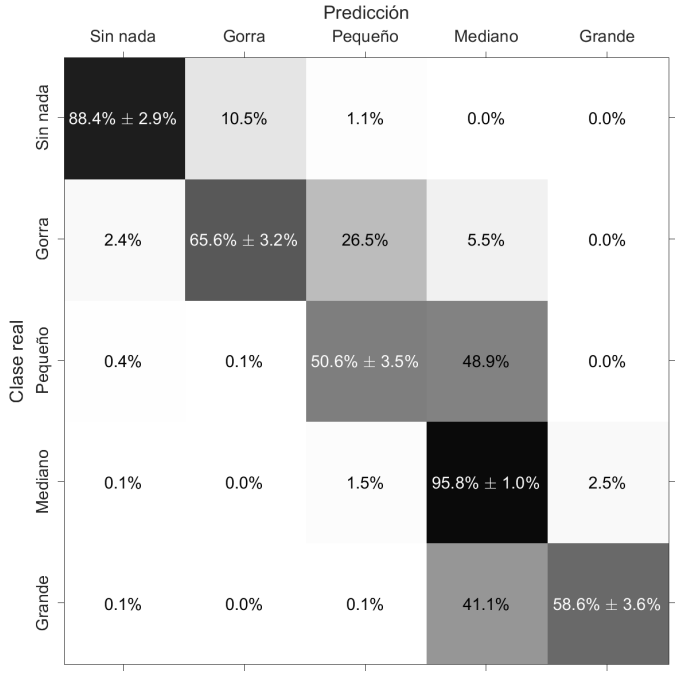


(b) Matriz de confusión obtenida utilizando la distancia de Mahalanobis para el cálculo del error de recuperación.

Figura 4.12: Resultados obtenidos para las secuencias de validación adquiridas con la cámara T0 para la clasificación en cinco clases de complementos.



(a) Matriz de confusión obtenida utilizando la distancia Euclídea para el cálculo del error de recuperación.



(b) Matriz de confusión obtenida utilizando la distancia de Mahalanobis para el cálculo del error de recuperación.

Figura 4.13: Resultados obtenidos para las secuencias de validación adquiridas con la cámara T1 para la clasificación en cinco clases de complementos.

4.3.5. Comentarios sobre los resultados

Según se observa en el apartado anterior 4.2. Los resultados obtenidos mediante el uso de la distancia euclídea para determinar el error de recuperación, demuestran un mayor porcentaje de

acierto frente a la distancia de Mahalanobis. Como conclusión para este tipo de complementos, los cuales tienen un tamaño relativamente pequeño es mas preciso usar la distancia euclidea.

También exponer que los resultados son independientes de la posición de la cámara, siendo estos similares al utilizar las dos distancias. Eso quiere decir que el sistema no se ve afectado por los cambios de angulo de grabación.

Comentar que, como cabe esperar, los porcentajes de éxito empeoran a medida que aumenta el número de clases, debido al incremento de la complejidad. En la primera matriz observamos tasas de aciertos muy elevadas (por encima del 90 %) 4.6. Según va aumentando las clase (mas elementos que debe distinguir) los resultados se ven afectados. Teniendo resultados muy positivos en las tres primeras matrices (2x2, 3x3 y 4x4) con tasas de aciertos superiores al 80 % . Es en la matriz de 5x5 donde se observa una caída mas brusca ya que la clasificación entre los sombreros de tamaño medio y pequeño es muy complicada debido a que su diferencia de tamaños son muy sutiles.

4.3.6. Estimación del coste computacional

Para que el sistema funcione en tiempo real se han de tener en cuenta dos cosas. La primera es la limitación que ofrece el ordenador, ya que la velocidad computacional es un factor importante. Y el otro requisito es que los algoritmos utilizados sean lo mas ágiles posibles.

Para este trabajo se ha utilizado un ordenador con las siguientes características. Un ordenador portátil HP Pavilion x360 con un procesador Intel(R)Core(TM) i7-7500 CPU@ 2,7GHz 2,9Ghz y 12GB de memoria RAM, sobre un sistema operativo Ubuntu 16.04.

Para comprobar que el sistema puede funcionar en tiempo real, se ha puesto una limitación en la cual un frame debe ser procesado en menos de treinta milisegundos. El procesado del frame se puede dividir en dos subprocesos, la localización del máximo correspondiente a la persona y la clasificación del complemento. La suma temporal de estos dos subprocesos no ha de ser mayor a treinta milisegundos.

En la siguiente tabla 4.6 se representan los valores medios para cada una de las etapas: detección de personas y clasificación de complementos, en función de los diferentes complementos considerados. Se puede observar que en todos los casos, el tiempo total es inferior a 30ms, permitiendo el funcionamiento del sistema en tiempo real. Además, las etapas añadidas en este TFG suponen un incremento muy pequeño del coste computacional, siendo la tarea más costosa la detección de personas.

	Sin Gorra/Sombrero	Gorra	Pequeño/Mediano	Grande
Tiempo de detección de máximos (seg)	0.01872	0.01709	0.02013	0.01975
Tiempo de clasificación de complementos (seg)	0.0058120	0.00567	0.00656	0.00634
Tiempo Total (seg)	0.02453	0.02276	0.02669	0.02609

Tabla 4.6: Relación de tiempos de procesación de cada clase.

Capítulo 5

Conclusiones y líneas futuras

En este capítulo se exponen las conclusiones obtenidas al observar los datos durante el estudio de dicho trabajo y se expondrán futuras lineas para mejorar e investigar que deriven de este proyecto.

5.1. Conclusiones

En el presente trabajo se ha implementado un sistema robusto para la clasificación de complemento. Las imágenes analizadas para extraer información son de profundidad. Dichas imagen ha sido extraída de dos cámaras ToF (Kinect II) ubicada en una posición cenital.

Dada una región de interés alrededor de una persona previamente detectada, la solución propuesta parte de trabajos emplea un descriptor de características basado en el número de medidas a diferentes alturas en la región correspondiente a la cabeza y hombres, y un clasificador basado en PCA.

Se han llevado acabo numerosas pruebas experimentales para la evaluación del sistema desarrollado. En ellas para verificar el ratio de acierto del sistema. Durante ellas se han utilizado distintos tipos de vídeos en los que hay diferentes ángulos de grabación, distintos individuos con complexiones físicas distintas y diferentes complementos. En todos los casos se han obtenido resultados que han permitido validar el sistema implementado en este TFG.

Para la obtención de resultados se han considerado cinco clases, organizadas de forma jerárquica de menor a mayor complejidad, obteniéndose los siguientes resultados:

- **Estudio de dos clases**: se engloban los resultados sin sombrero y con sombrero (gorra, sombrero grande,mediano y pequeño). Se obtiene que un 97,9 % de aciertos a la hora de comprobar si tiene sombrero y de un 92,3 % de aciertos al comprobar que no tiene sombrero. Esta es la clasificación mas sencilla y en la que mejores resultados se obtienen.
- **Estudio de tres clases**: en el siguiente paso, se desglosan mas las clases, generando una clasificación de: sin sombrero, con gorra y con sombrero (que incluye sombreros grandes, medianos y pequeños en una única clase). En este caso, los resultados son los siguientes: Sin sombrero: 92,3 %, Con gorra: 93,9 % y Con sombrero: 94,0 %.

- **Estudio de cuatro clases**: en este estudio se muestra una clasificación mas amplia y extensa en la que se comprueba los porcentajes de éxito del sistema dividiendo en Sin sombrero, Con gorra, Con sombrero pequeño o mediano, y Con sombrero grande dando los siguientes porcentajes: Sin sombrero: 92,3 %; Con gorra: 93, %; Con sombrero pequeño/-mediano: 82,4 %; Con sombrero grande: 86,1 %. De esta manera se comprueba que cuanto mas se particulariza la exactitud decae.
- **Estudio de cinco clases**: En el estudio de todas las clases los resultados son los siguientes: Sin sombrero: 92,3 %; Con gorra: 88,9 %; Con sombrero pequeño: 70,4 %; Con sombrero mediano: 68,3 %; Con sombrero grande: 86,1 %. Los aciertos durante este estudio son algo inferiores, sin embargo conviene destacar la complejidad del problema planteado.

Para la clasificación se han estudiado dos alternativas para el cálculo del error de recuperación de PCA: la distancia euclídea y la de Mahalanobis, obteniendo mejores resultados en el caso de la distancia Euclídea.

En conclusión el proceso de clasificación ha sido alcanzado con éxito. Aun así queda mucho trabajo de mejora y especialización.

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.

- Incorporar en el código lo necesario para conectarse a la cámara Kinect II. Este TFG cumple los límites establecidos para que funcione en tiempo real.
- Mejorar los tiempos de procesamiento. Esto puede ser posible de dos maneras. Usando algoritmos mas rápidos y eficientes o usando tecnologías mas avanzada. Mediante ordenadores que tengan una velocidad computacional mayor.
- Incluir en la clasificación tanto la información de profundidad como RGB. Dicha método debería mejorar la clasificación ya que tendría mas posibilidades a la hora de conseguir recursos que puedan ser usados para clasificar.

Bibliografía

[1] C. A. Luna, J. Macias-Guarasa, C. Losada-Gutierrez, M. Marron-Romera, M. Mazo, S. Luengo-Sanchez, and R. Macho-Pedroso, "Headgear accessories classification using an overhead depth sensor," *Sensors*, vol. 17, no. 8, p. 1845, 2017.

[2] "Cámara rgb," https://es.m.wikipedia.org/wiki/Archivo:Instax210wide.jpg[ultimo 31/07/2018].

[3] "Cámara tof," https://es.wikipedia.org/wiki/Archivo:TOF_Kamera.jpg[ultimo 31/07/2018].

[4] M. Hansard, S. Lee, O. Choi, and R. P. Horaud, *Time-of-flight cameras: principles, methods and applications*. Springer Science & Business Media, 2012.

[5] "Foto kinect ii," https://www.flickr.com/photos/bagogames/15232313609 [Útimo acceso julio 2018].

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

[7] C. S. Bamji, P. O'Connor, T. Elkhatib, S. Mehta, B. Thompson, L. A. Prather, D. Snow, O. C. Akkaya, A. Daniel, A. D. Payne *et al.*, "A 0.13 μm cmos system-on-chip for a 512$\times$ 424 time-of-flight image sensor with multi-frequency photo-demodulation up to 130 mhz and 2 gs/s adc," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 303–319, 2015.

[8] "Página del grupo de investigación geintra," http://www.geintra-uah.org/ [Útimo acceso julio 2018].

[9] D. Fuentes Jiménez, "Diseño, implementación y evaluación de un sistema de conteo de personas basado en cámaras de tiempo de vuelo," Trabajo Fin de Grado, Escuela Politécnica Superior. Universidad de Alcalá, 2016.

[10] R. García Jiménez, "Detección y conteo de personas, a partir de mapas de profundidad cenitales capturados con cámaras tof," Trabajo Fin de Grado, Escuela Politécnica Superior. Universidad de Alcalá, 2015.

[11] C. A. Luna, C. Losada-Gutierrez, D. Fuentes-Jimenez, A. Fernandez-Rincon, M. Mazo, and J. Macias-Guarasa, "Robust people detection using depth information from an overhead time-of-flight camera," *Expert Systems with Applications*, vol. 71, pp. 240 – 256, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417416306480

[12] "Información sobre ultra sonidos," http://picmania.garcia-cuervo.net/recursos/redpictutorials/sensores/sensores_de_distancias_con_ultrasonidos.pdf[ultimo 31/07/2018].

[13] R. García Jiménez *et al.*, "Detección y conteo de personas, a partir de mapas de profundidad cenitales capturados con cámaras tof," 2015.

[14] "Página web explicando descriptores del color," http://hpclab.ucentral.edu.co/wiki/index.php/Descriptores_de_color [Útimo acceso julio 2018].

[15] S. Palazuelos, L. M. Bergasa, M. Mazo, M. Ángel García, M. Escudero, and J. M. Miguel, "Apuntes va gitt: Representacion descripcion imagenes," Universidad de Alcalá, Tech. Rep., 2018.

[16] S. Hore, S. Chatterjee, S. Chakraborty, and R. K. Shaw, "Analysis of different feature description algorithm in object recognition," in *Feature Detectors and Motion Detection in Video Processing*. IGI Global, 2017, pp. 66–99.

[17] A. Fathi, P. Alirezazadeh, and F. Abdali-Mohammadi, "A new global-gabor-zernike feature descriptor and its application to face recognition," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 65–72, 2016.

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

[19] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli, "On the use of sift features for face authentication," in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*. IEEE, 2006, pp. 35–35.

[20] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[21] A. Jalal, Y.-H. Kim, Y.-J. Kim, S. Kamal, and D. Kim, "Robust human activity recognition from depth video using spatiotemporal multi-fused features," *Pattern recognition*, vol. 61, pp. 295–308, 2017.

[22] H. Choi and E. Kim, "New compact 3-dimensional shape descriptor for a depth camera in indoor environments," *Sensors*, vol. 17, no. 4, p. 876, 2017.

[23] N. A. Bakr and J. Crowley, "Histogram of oriented depth gradients for action recognition," *arXiv preprint arXiv:1801.09477*, 2018.

[24] G. Feng, Y. Liu, and Y. Liao, "Loind: An illumination and scale invariant rgb-d descriptor," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1893–1898.

[25] X. Xiao, S. He, Y. Guo, M. Lu, and J. Zhang, "Bag: A binary descriptor for rgb-d images combining appearance and geometric cues," in *International Conference on Cognitive Systems and Signal Processing*. Springer, 2016, pp. 65–73.

[26] S. Palazuelos, L. M. Bergasa, M. Mazo, M. Ángel García, M. Escudero, and J. M. Miguel, "Apuntes va gitt: Técnicas de machine learning para visión computacional," Universidad de Alcalá, Tech. Rep., 2018.

[27] S. Suthaharan, "Support vector machine," in *Machine learning models and algorithms for big data classification*. Springer, 2016, pp. 207–235.

[28] B. Yegnanarayana, *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

[29] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[30] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[31] M. T. E. Portillo and J. A. S. Plata, "P. ch. mahalanobis y las aplicaciones de su distancia estadistica," *CULCyT: Cultura Cientifica y Tecnologica*, no. 27, pp. 13–20, 2008.

[32] D. J. Cabello, "Sistema de monitorización mediante visión artificial de zonas de riesgo potencial para la circulación ferroviaria y seguridad de las personas," Trabajo Fin de Grado, Escuela Politécnica Superior. Universidad de Alcalá, 2009.

[33] "Página web de la base de datos gotpd1," http://www.geintra-uah.org/datasets/gotpd1 [Útimo acceso julio 2018].

[34] "Manual de instalación de opencv 12.4.9," https://www.samontab.com/web/2014/06/installing-opencv-2-4-9-in-ubuntu-14-04-lts/ [último acceso 05/agosto/2018].

[35] "Información sobre ubuntu," https://www.ubuntu.com/ [último acceso 1/noviembre/2013].

[36] "Eclipse for c/c++ developers," https://www.eclipse.org/callisto/c-dev.php [último acceso 05/agosto/2018].

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

[38] "Información sobre matlab," https://se.mathworks.com/products/matlab.html?requestedDomain= [último acceso 05/agosto/2018].

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, y se indica como utilizar y modificar el mismo. 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 16.04.1 LTS o superior.

A.2. Manual de usuario

A continuación se explica cómo instalar y utilizar el software implementado en este TFG. Así mismo, se proporciona información relacionada con la interpretación de los resultados proporcionados. Para ejecutar este proyecto son necesarios un conjunto de requisitos que se especifican en el anexo B.

Lo primero que se debe de hacer una vez se tiene el ordenador con los programas necesarios para hacerlo funcionar. Es instalar las librerias sobre las que se apoya el programa. En este caso las librerías son las de OpenCV (en concreto, se ha comprobado su funcionamiento con la versión 2.4.9). Para ello lo mejor es seguir los pasos de instalación de un manual [34].

Una vez que se han instalado todos los elementos necesarios se puede proceder a acceder al programa. Esto se puede hacer mediante dos caminos. Uno mediante terminal y el otro mediante una interfaz de ejecución de código como (Eclipse), cuyo entorno de trabajo se muestra en la figura A.1.

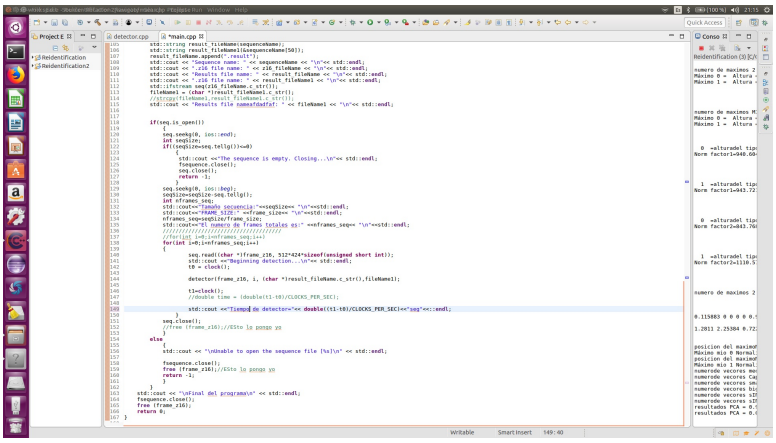


Figura A.1: Interfaz usada para trabajar en C++.

Para la ejecución del programa se requiere un fichero de texto plano, con extensión .list, en el que cada línea contiene el directorio donde se encuentran los vídeos a analizar. En la figura A.2 se muestra un ejemplo del contenido de un fichero .list.

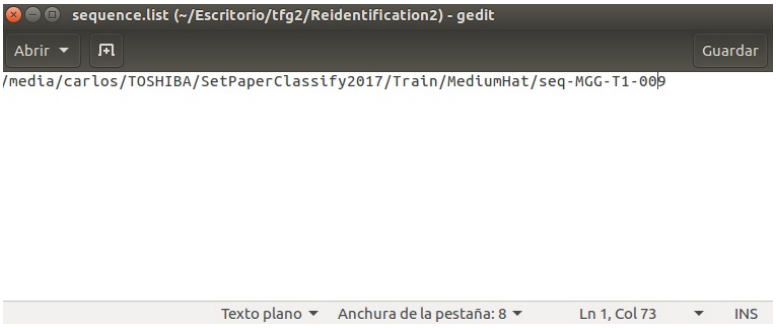
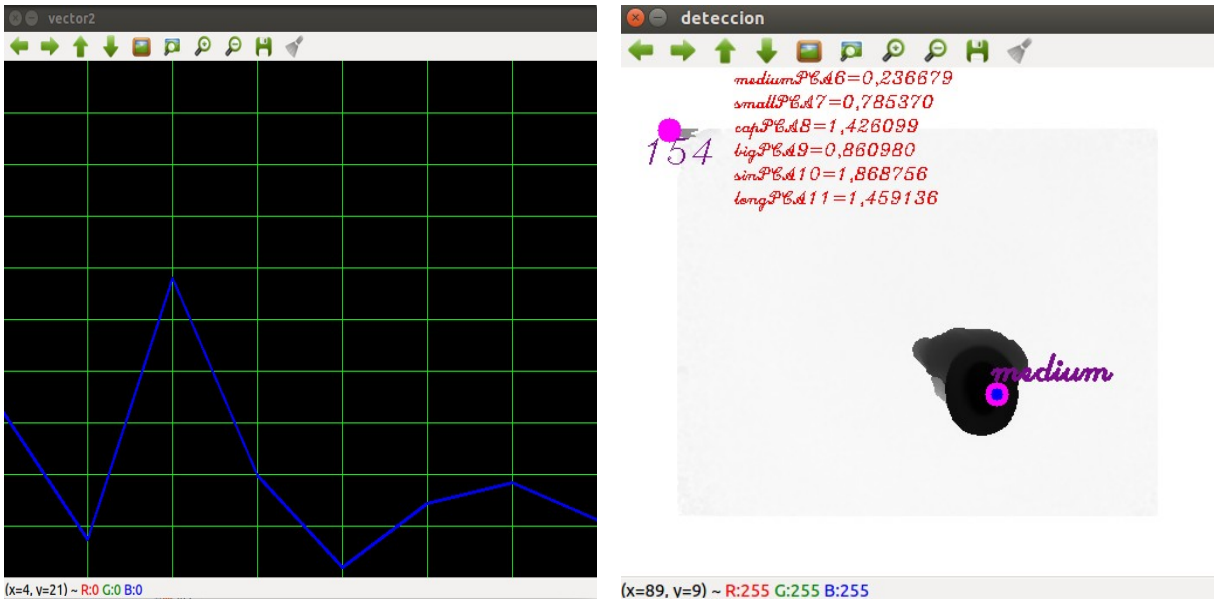


Figura A.2: Archivo .list donde se encuentra la ruta del video.

Una vez iniciado, el programa evalúa cada una de las imágenes incluidas en la carpeta que se ha indicado.

Una vez procesada la secuencia, el software proporciona dos secuencias de imágenes, y un fichero de texto con diferente información. En primer lugar se almacenan imágenes de los vectores de características (similares al ejemplo mostrado en la figura A.3a) obtenidos, además se guarda una imagen sobre la que se representan los máximos detectados, y para las personas, el tipo de complemento. Un ejemplo de la imagen de salida se muestra enla figura A.3b.



(a) Vector mostrado con información del elemento analizado.

(b) Imagen modificada mostrando informacion de los datos obtenidos.

Figura A.3: Elementos entregados por el software.

Además de las imágenes anteriores, se genera un archivo de texto plano, con extensión .dato, el cual incluye información sobre el tipo de elemento identificado, el frame en el que se ha localizado y la posición del frame en la que se ha localizado. Este tipo de archivo pueden ser de gran utilidad con un programa de postprocesado como (matlab) para generar matrices de confusión y otros elementos estadísticos.

En la siguiente figura A.4 se muestra el código el cual se usa para la extracción de datos.

```
void writesombrero(char* dato, char* nombre)
{
    //char vectorFileName[100];
      FILE *fp;
      char str1[10],str2[20];
      cout <<dato<< endl;
      cout <<"Size" <<sizeof(dato)<< endl;
      cout <<""<< endl;cout <<""<< endl;cout <<""<< endl;cout <<""<< endl;

      sprintf(str1,nombre);
      strcat(str1, ".dato");

     if ((fp = fopen(str1, "ab")) == NULL)
        {
         cout << "[ERROR] File " << nombre << " not open" << endl;
        }
    else
       {
        fputs(dato,fp);
        //fwrite (dato , sizeof(char), sizeof(dato)+4, fp);

        fputc('\n', fp);}
    fclose(fp);

}
```

Figura A.4: Código de extracción de información.

Durante la reproducción del programa es posible elegir la zona de trabajo, dicha zona es el espacio en la imagen en la que se realiza la detección de personas y clasificación de complementos. De esta manera las partes de la imagen que no interese se pueden obviar.

También se puede seleccionar el método que se desea utilizar a la hora de la clasificación: distancia de Mahalanobis o euclídea. En este manual se aconseja el uso de distancia euclídea, ya que ha sido la que ha proporcionado mejores resultados en los experimentos realizados.

El código mostrado en la figura A.5 corresponde a la sección del código fuente en el que se puede seleccionar el tipo de distancia a utilizar.

```
double PCAclasification2 (Mat samples, Mat actualCol, Mat cov)
{

    Mat actualRow = actualCol.t();
    Mat mean0, covar0, invcovar0;
    PCA pca(samples, Mat(), CV_PCA_DATA_AS_ROW, 3);
    Mat mean = pca.mean.clone();
    Mat eigenvalues = pca.eigenvalues.clone();
    Mat eigenvectors = pca.eigenvectors.clone();
    Mat imProject = pca.project(actualRow);
    Mat imReconstruct = pca.backProject(imProject);
    covar0 = cov/(samples.rows-1);

    invert(cov, invcovar0, DECOMP_SVD);
    invcovar0.convertTo(invcovar0,CV_32F);

    cout << "mean = " << invcovar0.cols<< endl;

    double normDifference = 0;
    double normDifference2=0;
    double normDifference3=0;

    Mat difference = Mat::zeros(1, NVECTOR, CV_32FC1);

    for (int i=0; i<NVECTOR; i++)
        difference.at<float>(0,i) = actualRow.at<float>(0,i) - imReconstruct.at<float>(0,i);


    for(int p=0;p<NVECTOR; p++)
        normDifference = normDifference + (difference.at<float>(0,p)*difference.at<float>(0,p));

    normDifference = sqrt(normDifference);

    normDifference3=Mahalanobis(imReconstruct,actualRow , invcovar0);

    cout <<"Diferencia Mahalanobis = " << normDifference3 << endl;
    cout <<"Diferencia Euclidea = " << normDifference << endl;


    return normDifference;
}
```

Figura A.5: Código de selección del tipo de medida de distancia a utilizar.

Apéndice B

Herramientas y recursos

Las herramientas necesarias para la elaboración del proyecto han sido:

- PC compatible
- 2 sensores de profundidad Kinect II [6, 7]
- Sistema operativo Ubuntu 16.04 LTS [35]
- Entorno de desarrollo Eclipse [36]
- Procesador de textos LaTeX[37]
- Software para el procesado de datos Matlab [38]

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 Hardware

- Procesador de 64 bits.
- Al menos 4 GB de RAM.
- Al menos 1 GB de memoria libres en el disco duro para funciones y datos.
- Al menos 20GB 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).
- Sensor Kinect II ubicado en posición cenital

C.2. Requisitos Software

- Sistema operativo Linux Ubuntu 16.04.1 LTS
- Libreria OpenCV 2.4.9
- Compilador GNU GCC

Apéndice D

Presupuesto

D.1. Costes de equipamiento

- Equipamiento Hardware empleado.

Concepto	Cantidad	Coste Unitario	Subtotal
Portátil táctil Convertible HP Pavilion x360	1	700€	700€
Kinect II	1	200€	200€
Coste total HW			900€

Tabla D.1: Costes del equipamiento Hardware empleado.

- Equipamiento software.

Concepto	Cantidad	Coste Unitario	Subtotal
Ubuntu	1	0€	€
Libreria OpenCV...	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	240 horas	60€/*hora*	14400€
Documentacion del TFG	60 horas	15€/*hora*	900€
Coste total			15300€

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

Concepto	Subtotal
Equipamiento HW	900€
Recursos software	0€
Mano de obra	15300€
Coste total del presupuesto	16200€

Tabla D.4: Coste total del presupuesto

D.3. Coste total del presupuesto

El importe total del presupusto asciende a la cantidad de : DIECISEISMIL DOSCIENTOS EUROS

En Madrid a ... de ..... de 2018.

Antonio Carlos Cob Parro.

Universidad de Alcalá
Escuela Politécnica Superior

ESCUELA POLITECNICA
SUPERIOR

Universidad
de Alcalá