Universidad Zaragoza
1542

Proyecto Fin de Carrera

Reconocimiento de acciones humanas en secuencias de vídeo

Autor

Pedro José Monforte Sánchez

Director

Carlos Orrite Uruñuela

Escuela de Ingeniería y Arquitectura

2012

Universidad Zaragoza

Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Proyecto Final de Carrera
Ingeniería de Telecomunicaciones
Curso 2011-2012

RECONOCIMIENTO DE ACCIONES HUMANAS EN SECUENCIAS DE VIDEO

Pedro José Monforte Sánchez

Mayo de 2012

Director: Carlos Orrite Uruñuela

Departamento de Ingeniería Electrónica y Comunicaciones
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

A mis padres y a Arantxa.

Resumen

En este proyecto se pretende conseguir el reconocimiento de acciones humanas en secuencias de vídeo. El tipo de acciones que se va a tratar consistirá en acciones simples ejecutadas por una sola persona en diferentes posiciones (por ejemplo, dar una patada, sentarse en el suelo, etc). El mayor problema que se abordará consistirá en el reconocimiento de estas acciones en situaciones de oclusión parcial de la figura, lo cual se produce en multitud de ocasiones en la vida real.

Trabajaremos con secuencias de vídeo de dominio público y libre acceso. Las secuencias de vídeo provienen de tres bases de datos de uso público que ya han sido utilizadas en estudios anteriores con un fin similar al nuestro, el reconocimiento de acciones humanas. Dichas secuencias podrán estar grabadas desde distintos puntos de vista con diferentes cámaras. Esto requiere un procesamiento previo de las imágenes para la extracción de características que se utilizarán en el clasificador. Es en este punto donde introducimos un nuevo descriptor ideado por nosotros basándonos en la dirección de los movimientos realizados en la ejecución de una acción. Además, nos encontramos con el problema de implementar un método de normalización de los datos de entrada al clasificador que sea independiente del grado de oclusión de la figura en la imagen.

Durante el desarrollo de este sistema de reconocimiento se emplean varias técnicas de procesado digital de imagen para la extracción de características. Además, el modelado de la acción humana se llevará a cabo mediante Modelos Ocultos de Markov (HMM), y su posterior reconocimiento se realizará también basándonos en dicha metodología.

Agradecimientos

A Gañán, Velasco, Rossi, Murphy y otros compañeros de clase con los que he compartido demasiadas horas de estudio, prácticas y cafetería; tantas que no podría contarlas.

A Carlos Orrite, por la disposición e interés puestas en este proyecto y esas ideas que lo han llevado a buen destino. Gracias por darme la oportunidad de hacer un proyecto interesante como este.

A mis compañeros de laboratorio con los que he compartido días y días. A los Migueles, por su inestimable ayuda y por esas charlas y cafés que hacían más cortas las jornadas delante del ordenador. Especialmente a Mario, que ha estado estos últimos meses aguantando mis dudas y ha echado en este proyecto más tiempo del que disponía.

A mi familia, por todos esos ánimos cuando todo se venía encima. Teníais razón; algún día tenía que terminar.

En definitiva, a todos aquellos que han hecho posible que terminase de la mejor manera. Muchas Gracias a todos.

Índice general

Capítulo 1

Introducción

El reconocimiento automático de acciones humanas es actualmente uno de los campos de investigación más activos en el área de la visión por computador. Esto es debido en gran medida al creciente interés en la interpretación automática de escenas de vídeo en potenciales aplicaciones como: *interacción hombre-máquina*, para fines sociales o para la industria del entretenimiento; *vídeo vigilancia*, donde el sistema detecta, reconoce y actúa ante acciones u objetos sospechosos; o *análisis del rendimiento deportivo*, analizando automáticamente las acciones de los atletas con el objetivo de proporcionar asistencia en la mejora de su realización. Sin embargo, el reconocimiento de acciones es un problema complejo debido a los numerosos factores implicados: desde la gran diversidad existente entre las personas tanto en su apariencia como en el estilo de ejecución de la acción, pasando por el escenario donde se llevan a cabo, afectados por sombras, cambios de iluminación u oclusiones e incluyendo otros factores, como el ángulo de vista y la distancia del sujeto respecto a la cámara. Además, las acciones humanas llevan asociadas una componente espacial y una componente temporal, ambas altamente afectadas por los factores anteriores, lo cual implica una alta aleatoriedad en su ejecución.

Existen, por tanto, numerosas técnicas que se aplican a esta tarea: bien basadas en la definición de un modelo del cuerpo humano (2D o 3D) o en las características extraídas directamente de las muestras de vídeo; bien, usando información sobre la forma del sujeto o sobre su patrón temporal. Así, los métodos investigados en el análisis de las acciones humanas independientes de la vista se pueden clasificar en dos categorías: los basados en comparación de patrones y los que se basan en modelos de espacios de estados [14]. Como veremos a lo largo de este trabajo, el método empleado se basa en una clasificación mediante un modelo de estados, utilizando un nuevo descriptor de características independiente de dicho modelo.

1.1. Descripción del proyecto

Este proyecto de fin de carrera ha sido realizado en el Laboratorio de Visión por Computador, grupo de investigación perteneciente al departamento de Ingeniería Electrónica y Comunicaciones de la Escuela de Ingeniería y Arquitectura (EINA) de la Universidad de Zaragoza. Dicho trabajo se ha llevado a cabo bajo la dirección del Dr.Ing. D.Carlos Orrite Uruñuela.

En este proyecto se desarrolla un sistema de reconocimiento de acciones humanas en secuencias de vídeo de forma independiente a la perspectiva. El tipo de acciones que se tratarán consistirá en acciones simples ejecutadas por una sola persona en diferentes posiciones (por ejemplo dar una patada, caminar, sentarse, etc.). Dichas acciones son realizadas en situaciones sin y con oclusiones parciales del sujeto y podrán estar grabadas desde distintos puntos de vista con diferentes cámaras. Trabajaremos con bases de datos públicas y de libre acceso usadas en diversos trabajos sobre el reconocimiento de acciones (*ver anexo A*). La intención es que nuestro sistema consiga diferenciar las acciones aunque en la secuencia de vídeo se encuentren regiones ocultas durante su ejecución. El trabajo abarca todas las fases de un sistema de reconocimiento: pre-procesamiento de las imágenes, extracción de características y clasificación.

1. *Pre-procesado de las imágenes.* Consiste en la eliminación de posible ruido y la mejora de la calidad de las imágenes mediante el uso de filtros digitales. En este paso se lleva a cabo también la segmentación de los objetos de interés, es decir, los sujetos.

2. *Extracción de características.* Se busca el conjunto de datos representativos que describen las acciones para su posterior clasificación en distintas clases.

3. *Clasificación.* Se emplean las características extraídas para asignar cada acción a la clase más apropiada según el clasificador.

A continuación se describe el método desarrollado en este proyecto:

En primer lugar, se emplea la *Motion History Image* (MHI) [3] como modelo de representación del movimiento por su robustez demostrada en investigaciones de reconocimiento de acciones y análisis del movimiento [2, 3, 4, 7, 8]. Este es un modelo temporal donde se representa la acción según los movimientos realizados en la misma a lo largo del tiempo y condensa esta información en una sola imagen, donde la intensidad de cada píxel es función de la historia de la actividad en dicha localización dentro de la secuencia de imágenes.

A partir de este modelo se puede obtener la orientación del movimiento global por medio del cálculo de los gradientes de intensidad sobre la MHI [4]. Ya que la intensidad de la MHI es función de la actividad temporal podemos calcular los gradientes de intensidad mediante una convolución con filtros Sobel separables en las direcciones vertical y horizontal y estos nos darán una aproximación de la orientación del movimiento dentro de la secuencia de vídeo analizada.

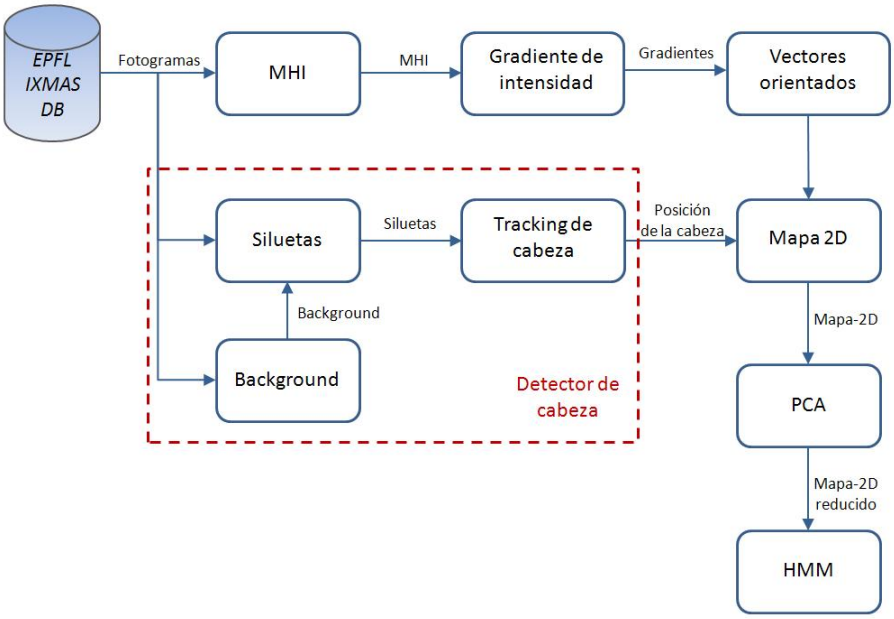


Figura 1.1: *Diagrama de bloques del sistema para secuencias con oclusiones.*

Después, la idea es obtener algún parámetro normalizado que sea capaz de caracterizar las distintas acciones, de forma que sirva posteriormente en la etapa de clasificación. Hemos optado por probar un nuevo descriptor a partir de las orientaciones de los gradientes de intensidad. Se trata de representar las acciones mediante un histograma bidimensional de gradientes donde los vectores de gradiente se contabilizan según su orientación y su localización angular respecto de la cabeza del sujeto, como se explica más detalladamente en la sección 4.2 de esta memoria. Para ello se requiere estimar la posición de la cabeza, lo que se logra mediante un bloque de detección y seguimiento (*figuras 1.1 y 1.2*). Estos histogramas de gradientes serán nuestros vectores de características. A menudo nos referiremos a ellos a lo largo de la memoria como *mapas-2D*.

Estos vectores de características se usan para la clasificación de las acciones mediante un clasificador probabilístico basado en Modelos Ocultos de Markov o HMM (*ver anexo B*). Como

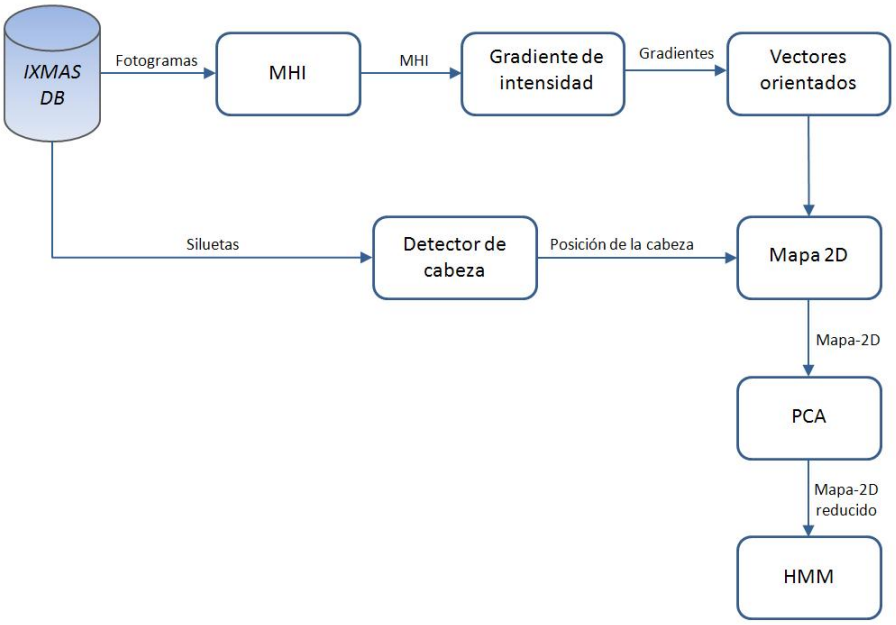


Figura 1.2: *Diagrama de bloques del sistema para secuencias sin oclusiones.*

se explica en la sección 5.2.2, este clasificador es entrenado con una base de datos en la que las secuencias no presentan oclusiones. Posteriormente, este clasificador se utilizará en la clasificación de dos bases de datos distintas a la primera. Estas otras dos bases también difieren entre sí: una contiene las secuencias de entrenamiento a las que se han añadido oclusiones artificiales, mientras la otra contiene nuevas secuencias sin y con oclusiones (*ver anexo A*). Se medirá la eficiencia mediante la técnica de validación *leave-one-subject-out*, donde dejamos las secuencias correspondientes a un sujeto como control mientras se entrena el clasificador con el resto de secuencias; repitiendo el proceso dejando como control en cada iteración a cada uno de los sujetos y promediando el resultado de todos. Al final, se compararan los resultados obtenidos con el mismo clasificador para las distintas bases de datos.

1.2. Objetivos del proyecto

El objetivo de este proyecto es el reconocimiento de acciones humanas en secuencias de vídeo. El principal problema que abordaremos consistirá en el reconocimiento de estas acciones en situaciones de oclusión parcial de la figura, lo cual se produce en multitud de ocasiones en la vida real. Este proyecto abarca el desarrollo de un sistema de reconocimiento para tal fin; partiendo del pre-procesado de las secuencias de vídeo, la extracción de las características que

modelan la acción humana y su posterior reconocimiento.

Para lograr los retos que supone este proyecto, se plantean los siguientes objetivos parciales:

1. Identificar bases de datos de acciones humanas. Inicialmente se llevará a cabo una documentación de las bases de datos que se ajustan a las características de nuestro proyecto (distintas perspectivas de la acción, presencia de oclusiones, acciones individuales, etc.).
2. Analizar la metodología más adecuada para caracterizar la figura humana en situaciones de oclusión parcial de la misma. Una vez elegidas las secuencias con las que trabajaremos, se pasará al procesado de las imágenes para conseguir el modelado de la acción.
3. Implementar un método de normalización de los datos de entrada al clasificador, logrando que las características del modelo a clasificar sea independiente del grado de oclusión de la figura en la imagen.
4. Modelar la acción de una forma invariante al punto de vista de la cámara. Ello implicará entrenar el clasificador con diferentes modelos correspondientes a varias posiciones del cuerpo respecto de la cámara.

1.3. Organización de la memoria

La memoria está estructurada en 7 capítulos y 2 anexos. A continuación se describen los capítulos restantes:

- *Capítulo 2: Generación de las siluetas.*
  Se detalla el procesado de las imágenes realizado en la extracción de las siluetas humanas que serán usadas por el detector de cabeza.
- *Capítulo 3: Modelado del movimiento.*
  Descripción de los algoritmos empleados para la representación de la acción y la estimación de la dirección del movimiento.
- *Capítulo 4: Extracción de características.*
  En este capítulo se detalla el proceso de generación del vector de características usado en la clasificación. Además, se describe el método seguido en la detección de la cabeza a partir de las siluetas.

- *Capítulo 5: Diseño del clasificador.*
  Se presenta el clasificador empleado para el reconocimiento y se detalla la elección de los parámetros del mismo.

- *Capítulo 6: Análisis de los resultados.*
  Se presentan los resultados de reconocimiento obtenidos con el clasificador diseñado, así como las pruebas llevadas a cabo y la descripción de los grupos de datos empleados para las mismas.

- *Capítulo 7: Conclusiones y trabajo futuro.*
  En el capítulo final se valorará el cumplimiento de los objetivos definidos anteriormente y se sugieren modificaciones en algunos aspectos que podrían mejorarse para un futuro.

- *Anexo A: Bases de datos.*
  Documentación de las bases de datos para reconocimiento de acciones de donde provienen las secuencias de vídeo utilizadas en este proyecto.

- *Anexo B: Modelos ocultos de Markov.*
  Se explica la base teórica de los modelos ocultos de Markov empleados en el diseño del clasificador del sistema de reconocimiento desarrollado en este proyecto.

Capítulo 2

Generación de siluetas

Una de las técnicas básicas en el análisis de secuencias de vídeo es la de diferenciar o separar del fondo de la escena (*background*) los objetos en movimiento ajenos a él (*foreground*). Esta operación suele conocerse como segmentación. Mediante este proceso se asigna una etiqueta, en este caso *background* o *foreground*, a cada píxel de la imagen de forma que los píxeles que compartan la misma etiqueta son similares en alguna característica, como el color, la intensidad o la textura. Existen diversas técnicas de segmentación de secuencias de vídeo dependiendo de los objetivos perseguidos y la naturaleza de las secuencias. En escenas grabadas por una cámara fija, como es el caso que se da en este trabajo, las técnicas de segmentación más eficaces son las basadas en el modelado del fondo y su posterior sustracción (*figura 2.1*). El modelado de fondo tiene como objetivo la estimación del fondo a partir de la secuencia de vídeo y, la sustracción de fondo es el proceso mediante el cual cada fotograma de la secuencia se compara con el modelo de fondo para identificar los píxeles que pertenecen a nuestro *foreground*, es decir, la silueta del sujeto. La extracción del fondo de estas siluetas se calcula mediante la diferencia de ambas imágenes. Estas diferencias serán umbralizadas para obtener las siluetas de los sujetos mediante una segmentación binaria de la imagen. Debido a que las secuencias de vídeo no son ideales (pueden presentar cambios de iluminación, sombras, reflejos, ruido de la cámara, etc.), la extracción de las siluetas presenta múltiples complicaciones. Para minimizar los errores y mejorar la forma, se realiza un post-procesado de las siluetas binarias mediante filtros espaciales y morfológicos. Estas siluetas serán la información que se le suministre al detector de cabeza que se detalla en la sección 4.1.

Hay que destacar que los métodos de generación de siluetas que se detallan aquí no son métodos generalizables, sino que son específicos para las bases de datos[1] empleadas en este trabajo: IXMAS (INRIA Xmas Motion Acquisition Sequences) Dataset, IXMAS con oclusiones artificiales (O-IXMAS) y EPFL-IXMAS (École Polytechnique Fédérale de Lausanne). Nuestro interés principal es la localización de la cabeza, suponiendo conocidas las siluetas. Sin embargo, mientras que la base de datos IXMAS es la más completa (contiene las siluetas binarias para cada fotograma de las secuencias y el fondo del escenario de la grabación), en las bases de datos O-IXMAS Y EPFL-IXMAS no se dispone de siluetas ni imágenes del escenario. Por tanto, en este proyecto se ha intentado optimizar su obtención a partir de la información concreta de estas bases de datos para, posteriormente, aplicar un detector de cabeza basado en las imágenes de las siluetas. Por tanto, el bloque de extracción de siluetas es previo al detector de cabeza al trabajar con las bases de datos O-IXMAS y EPFL-IXMAS.

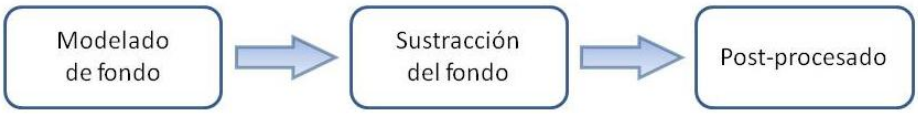


Figura 2.1: *Diagrama de bloques de la generación de las siluetas.*

2.1. Modelado de fondo

El problema del modelado de fondo se complica por el hecho de que tampoco se proporcionan imágenes del fondo de la escena en las bases de datos O-IXMAS y EPFL-IXMAS. Este problema es distinto para cada una de las dos bases de datos porque las características de las secuencias de vídeo son distintas. Mientras que en la base de datos EPFL-IXMAS el escenario con oclusiones varía para cada secuencia, en la base de datos O-IXMAS se introducen oclusiones artificiales que cambian para cada secuencia de cada acción, por lo que el método de modelado de fondo empleado en cada una será distinto. Para las secuencias de EPFL-IXMAS se realiza un filtrado de mediana temporal sobre un fragmento de la propia secuencia de vídeo, como se explica en la sección 2.1.1. En cambio, como las secuencias disponibles en O-IXMAS son una modificación de las que se encuentran en la base de datos IXMAS, se aprovechan las imágenes del fondo de esta última para el filtrado de mediana, proceso que se detalla en la sección 2.1.2.

[1] El contenido y la información de las bases de datos se detalla en el *anexo A*.

2.1.1. Filtro de mediana

Una aproximación para modelar el fondo de una escena es suponer que los píxeles del fondo no cambian de valor a lo largo de la secuencia; es decir, sólo el movimiento de los sujetos producirá variación en la intensidad de estos píxeles. Esta suposición se suele realizar en entornos muy controlados; por ejemplo, interiores con una imagen de fondo estática y sin variaciones de iluminación como los empleados en este proyecto. Uno de los métodos más utilizados para la estimación de fondo estático es el filtro temporal de mediana [9, 13, 18, 21]. Este filtro almacena de manera ordenada los valores de los píxeles de un conjunto de N imágenes y calcula el fondo, I_B, como la mediana de los valores almacenados para cada píxel (*ecuación 2.1*).

$$I_B(x,y) = med\{I_N(x,y), I_{N-1}(x,y), \ldots, I_1(x,y)\} \tag{2.1}$$

En este método es necesario que el fondo esté visible la mayor parte del tiempo para eliminar completamente los elementos del frente o *foreground*. Por ello, el conjunto de imágenes empleado en el modelado del fondo para las secuencias contenidas en la base de datos EPFL-IXMAS es el que corresponde a la grabación del sujeto agachándose (*acción 4*). Estas secuencias tienen una duración que varía entre 2.30 y 3.87 segundos (N=$[53, 89]$ fotogramas) con una media de 3.05 segundos (N=70 fotogramas). Puesto que el escenario con objetos varía para cada secuencia, también se necesita generar una imagen del fondo para cada una. Aunque se probó a realizar el filtrado de mediana con otras acciones y combinaciones de ambas, se ha comprobado experimentalmente que ésta ofrece mejores resultados para nuestro propósito de generar siluetas útiles en la detección de la cabeza. El método empleado es capaz de modelar el fondo del área por la que se desplaza la cabeza a excepción de la región inferior de la imagen, donde la sombra o las piernas del sujeto producen errores en la estimación.

2.1.2. Ajuste de la bounding box

El método del apartado anterior no se puede aplicar a la base de datos O-IXMAS, ya que contiene oclusiones artificiales que son diferentes entre sujetos y, ademas, entre las acciones realizadas dentro de una misma secuencia. Por tanto, es necesario generar un fondo distinto para cada acción.

En la documentación de la base de datos disponible en internet [23] se especifica que para esta base de datos se utiliza una región de interés (ROI) estática de tamaño 48x64 píxeles centrada en el sujeto. Como esta base de datos es una modificación de la base de datos IXMAS, en la cual disponemos del fondo, podemos ajustar una región alrededor del sujeto y tomar esta zona de la imagen de fondo para generar las siluetas. Para encontrar esta región de dimensiones desconocidas, primero se localiza el centroide del sujeto a partir de las siluetas disponibles en IXMAS. Se generan una serie de ROIs en la imagen del fondo disponible en la base IXMAS centradas en el centroide calculado y con dimensiones que son múltiplos de 48x64 porque se desconoce el escalado utilizado en esta base de datos respecto de las imágenes de IXMAS. Estas ROIs se reducen al tamaño original de 48x64 y se realiza la diferencia entre las ROIs del fondo y las imágenes de la base O-IXMAS que se procesan, eligiendo como fondo aquella ROI que minimice esta diferencia. Finalmente, el fondo se calcula aplicando un filtro de mediana como el del apartado anterior sobre las ROIs del fondo de la base IXMAS, que después del escalado tendrán unas dimensiones de 48x64 píxeles.

2.2. Extracción del fondo

La extracción o sustracción de fondo es una técnica comúnmente usada para segmentar los objetos de interés de la escena. Se lleva a cabo comparando la imagen procesada con una imagen de fondo estimada y aquellas áreas donde haya una diferencia significativa entre ambas indicarán la localización y la forma de los objetos de interés. La comparación entre las dos imágenes se realiza mediante su diferencia píxel a píxel. Los objetos que se diferencian del fondo producirán valores distintos de cero y, después, estos valores serán umbralizados para eliminar posibles diferencias en la captura de las imágenes (ruido) y segmentar dichos objetos. Así, a partir de una imagen de fondo I_B y una imagen a analizar I_t, se obtiene la silueta en el instante t ,S_t, como:

$$S_t(x,y) = \begin{cases} 1, & \text{si } |I_B(x,y) - I_t(x,y)| > th, \\ 0, & \text{en otro caso.} \end{cases} \tag{2.2}$$

En la ecuación 2.2 se realiza la diferencia absoluta de los niveles de gris de los píxeles de la imagen del fondo I_B y de la imagen en el instante t, I_t. Esta diferencia es umbralizada con el valor th para obtener una imagen binaria. Trabajaremos con imágenes en escala de grises, con valores entre 0 y 255, porque sólo disponemos de esta calidad de vídeo en las bases de datos.

Así, el umbral se elige experimentalmente como th=25, es decir, los píxeles de la imagen I_t que se diferencian de los píxeles del fondo en más de 25 niveles de gris serán considerados como parte del sujeto.

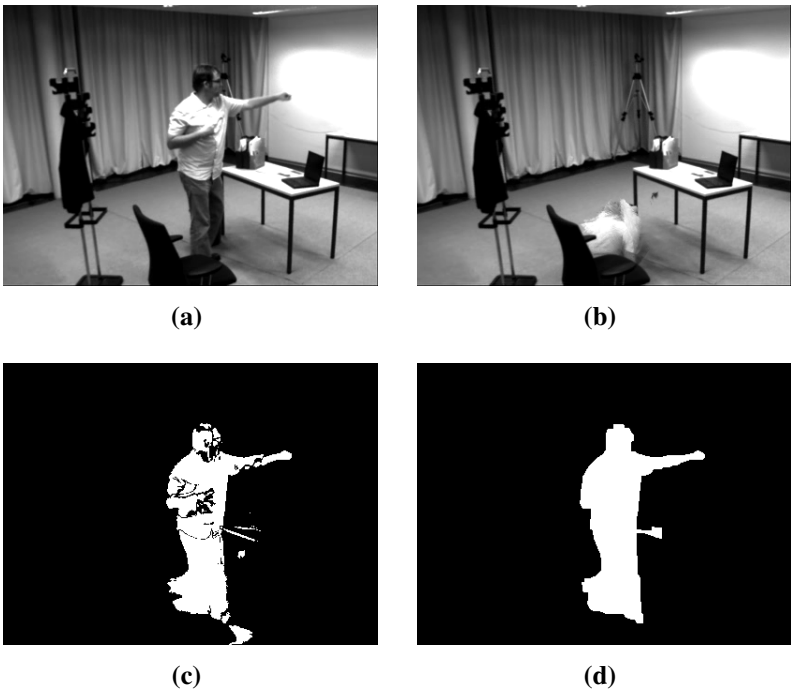


Figura 2.2: *(a) Imagen original de la base de datos EPFL-IXMAS. (b) Fondo estimado para esa secuencia de vídeo. (c) Resultado de la extracción de fondo mediante la ecuación 2.2. (d) Silueta resultante después del post-procesado.*

Esta silueta que hemos calculado puede contener errores debido a cambios de iluminación en la escena, ruido de la cámara o errores derivados de la estimación del fondo. Para minimizar estos errores se realiza posteriormente un procesado de las siluetas mediante filtros morfólogicos binarios de apertura y cierre y filtros espaciales de mediana. Los filtros espaciales eliminan píxeles aislados, mientras que con el filtro de apertura se eliminan salientes finos y se suaviza el contorno de la silueta y con el filtro de cierre se eliminan pequeños huecos y se unen cortes finos en la silueta. Estos filtros morfológicos son combinaciones sucesivas de operaciones de dilatación y erosión [10] con un elemento estructural de 3x3 píxeles de valor 1. Para rellenar posibles huecos en el interior de las siluetas se les aplican algoritmos de procesamiento de imágenes para el relleno de regiones en imágenes binarias. En la figura 2.2 se pueden ver las imágenes resultantes después de cada paso. Las imágenes 2.2a y 2.2b se corresponden a una imagen original de la base de datos EPFL-IXMAS y la imagen de fondo estimada con el filtro de la sección 2.1.1.

A partir de estas dos imágenes se obtiene la silueta de la imagen 2.2c mediante la extracción de fondo de la ecuación 2.2. Después de aplicar una serie de filtros morfológicos obtenemos nuestra silueta final (*figura 2.2d*).

Capítulo 3

Modelado del movimiento

En un sistema de reconocimiento de acciones como éste la mayor dificultad se presenta en la extracción de características. En este capítulo se detalla el proceso seguido para modelar el movimiento presente en las secuencias de vídeo analizadas, generando los patrones espacio-temporales que representan la ejecución de las acciones de forma compacta. Como se verá más adelante en este capítulo, por medio del cálculo de vectores de gradiente sobre estos patrones podemos extraer la información de la orientación del movimiento realizado durante la ejecución de la acción.

3.1. Representación de la acción

La representación de los movimientos realizados al ejecutar una acción dentro de la secuencia de vídeo se ha llevado a cabo mediante el método MHI (*Motion History Image*). Este método es una herramienta ampliamente utilizada en el análisis del movimiento humano [2, 3, 4, 7, 8], donde se convierte una secuencia de imágenes en un modelo estático que condensa la información espacio-temporal del movimiento presente en ella. Este enfoque, aunque es más sensible al ruido y a la variabilidad de los movimientos ejecutados en un intervalo temporal, requiere una menor carga computacional que los basados en modelos de estados o en descriptores semánticos.

3.1.1. Motion History Image

La imagen de la historia del movimiento o *Motion History Image* (MHI) es un método basado en plantillas temporales donde el movimiento presente en una secuencia de imágenes se condensa en imágenes en escala de grises. De esta manera se puede representar el movimiento de forma compacta. La MHI captura la información del movimiento presente en los fotogramas de la secuencia de vídeo en una sola imagen estática donde el valor de cada píxel es función de la historia temporal del movimiento en esa localización espacial de la imagen.

Como se explica en [3], la imagen MHI puede ser calculada a partir de una función de actualización:

$$MHI_t(x,y) = \begin{cases} \tau, & \text{si } \Psi(x,y,t) = 1, \\ \text{máx}(0, MHI_{t-1}(x,y) - 1), & \text{en otro caso.} \end{cases} \tag{3.1}$$

Donde (*x*,*y*) muestran la posición espacial, *t* es el número del fotograma actual dentro de la secuencia analizada y τ es el número de fotogramas que componen dicha secuencia, cuya información es condensada en la MHI. La función de actualización ,$\Psi(x,y,t)$, señala la presencia de movimiento en la imagen actual y se calcula para cada nuevo fotograma analizado en la secuencia. El resultado de estos cálculos es una imagen de valores escalares donde los píxeles con movimiento más reciente son los más brillantes, mientras que los movimientos más antiguos se corresponden con píxeles de menor intensidad.

La función de actualización lleva a cabo la segmentación del movimiento presente en la imagen y se puede definir mediante algunas técnicas de procesamiento de imagen como la sustracción del fondo, la diferencia de imágenes o el flujo óptico. Normalmente, la MHI es generada partiendo de imágenes binarias obtenidas de la diferencia de fotogramas y su posterior umbralización (*ecuaciones 3.2 y 3.3*). En este proyecto se ha optado por un enfoque de ese estilo, basado en la diferencia umbralizada de fotogramas consecutivos, el cual sigue siendo un método robusto para detectar de una forma sencilla la presencia de movimiento.

$$D(x,y,t) = |I(x,y,t) - I(x,y,t-1)| \tag{3.2}$$

donde $I(x,y,t)$ es el valor de intensidad del píxel localizado en (x,y) en el fotograma *t* de la secuencia de vídeo.

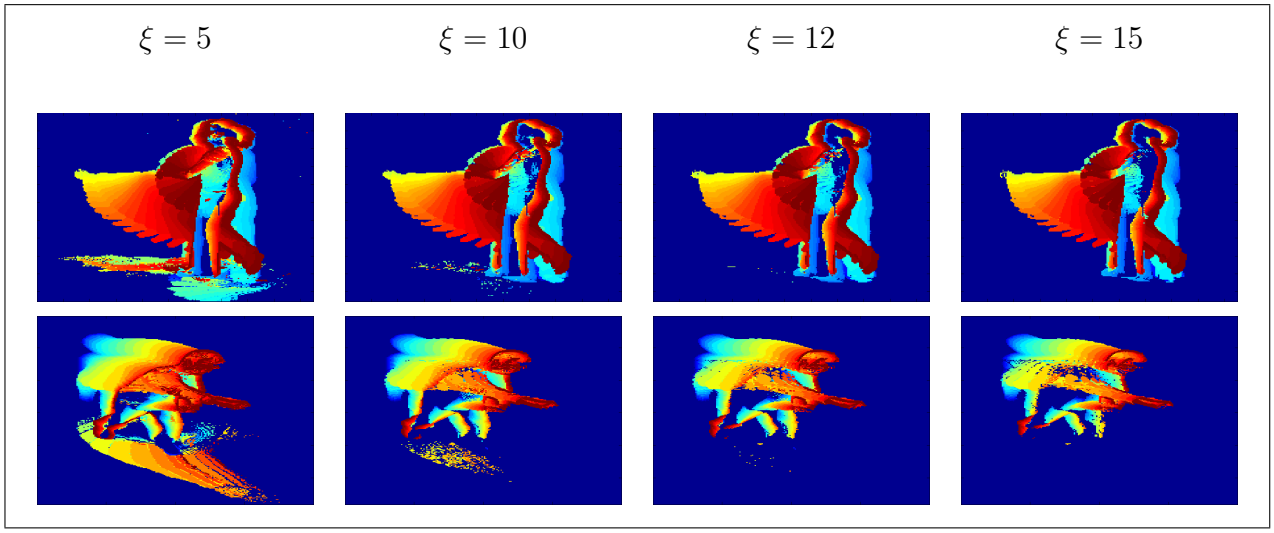


Figura 3.1: *MHI obtenidos variando el valor del umbral para diferentes acciones: dar una patada (fila superior) y dar un puñetazo (fila inferior).*

Las secuencias de vídeo analizadas tienen fondo estático; aun así cambios puntuales en ese fondo, debidos a cambios en la iluminación, ruido en la captura de las imágenes o a las sombras dinámicas provenientes del sujeto, son algunos de los factores que pueden dificultar la segmentación del movimiento. Por ello, a las diferencias absolutas entre niveles de gris de dos imágenes consecutivas de la ecuación 3.2 se les aplica un umbral, ξ, obtenido experimentalmente (*figura 3.1*). Podemos obtener finalmente la plantilla MHI mediante la combinación de las ecuaciones 3.1 y 3.3.

$$\Psi(x,y,t) = \begin{cases} 1, & \text{si } D(x,y,t) \geq \xi, \\ 0, & \text{en otro caso.} \end{cases} \tag{3.3}$$

3.1.2. Segmentación de la acción

Uno de los problemas clave de la MHI es la auto-oclusión de la acción, donde movimientos recientes enmascaran la información de movimientos anteriores en la misma región de la imagen durante la ejecución completa de una acción. Si la realización de una acción se compone de movimientos en direcciones opuestas, como por ejemplo levantar y bajar el brazo para volver al reposo, entonces la información del movimiento pasado se pierde al superponerse a éste otro movimiento más reciente. Otro inconveniente de este método es que, como cualquier método basado en plantillas, es sensible a la variación de la velocidad de ejecución de los movimientos.

Teniendo en cuenta la variabilidad de la duración de las acciones y el problema de la auto-oclusión hemos optado por dividir cada secuencia de ejecución de una acción en un número fijo de segmentos solapados al 50 %. Estos segmentos serán las ventanas sobre las que calcularemos cada MHI (*figura 3.2*), de forma que cada acción se constituye de sucesivas MHI que se corresponden con dichos segmentos. El número de segmentos en que dividimos las secuencias se ha tomado como un parámetro de diseño, tomando 5 y 10 ventanas por cada acción; comparando los resultados obtenidos en el entrenamiento del clasificador para cada uno de estos valores. Estos resultados se podrán ver en la sección 5.2.2 de esta memoria.

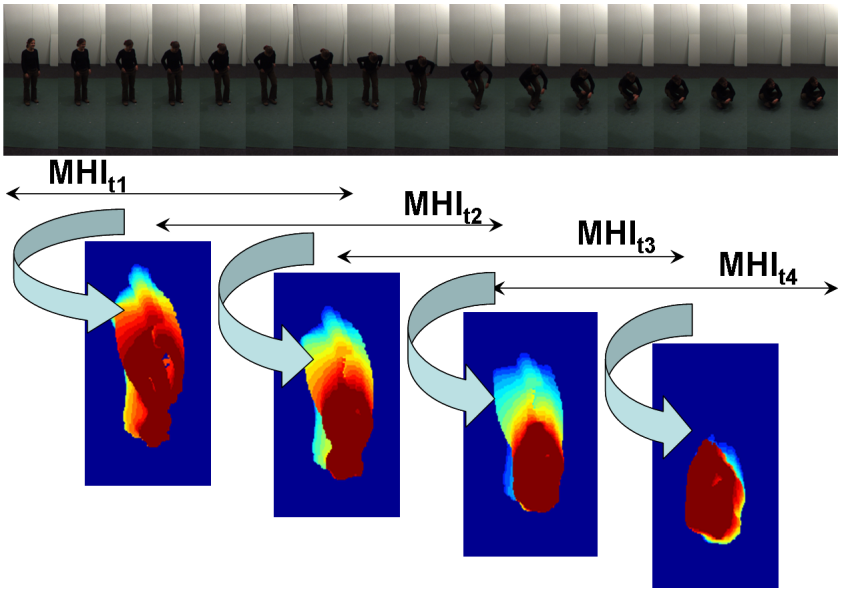


Figura 3.2: *Segmentos solapados y sus correspondientes MHI.*

3.1.3. Post-procesado de la MHI

Las imágenes MHI calculadas según la ecuación 3.1 pueden contener píxeles ruidosos o huecos dentro de la región principal del movimiento derivados de las diferencias realizadas entre fotogramas consecutivos (*ecuación 3.2*). Así que previamente al cálculo de gradientes, que se verán en posteriores secciones de este capítulo, se realiza un post-procesado mediante la técnica de procesamiento morfológico de imágenes conocida como suavizado morfológico (*morphological smoothing*). Primeramente la MHI se pasa por un filtro morfológico de apertura-cierre (*open-closing*) que consiste, como su nombre indica, en la concatenación de una operación morfológica de apertura y una de cierre. Ambas operaciones se basan a su vez en operaciones básicas de dilatación y erosión de imágenes. Así, la operación morfológica de apertura de una imagen I por un elemento estructural B se define como [10]:

$$I \circ B = (I \ominus B) \oplus B \tag{3.4}$$

Lo que puede verse como una operación de erosión (*ecuación 3.5*) seguida de una operación de dilatación (*ecuación 3.6*), usando ambas el mismo elemento estructural B. El elemento estructural utilizado es una matriz de tamaño 3x3 siendo todos sus elementos la unidad. Esta matriz puede interpretarse como una máscara de convolución a la hora de aplicar estos filtros a la imagen.

$$[I \ominus B](x, y) = \text{mín}\{I(x + s, y + t)|(s, t) \in B\} \tag{3.5}$$

$$[I \oplus B](x, y) = \text{máx}\{I(x - s, y - t)|(s, t) \in B\} \tag{3.6}$$

A su vez la operación morfológica de cierre de una imagen I por un elemento estructural B está definida como [10]:

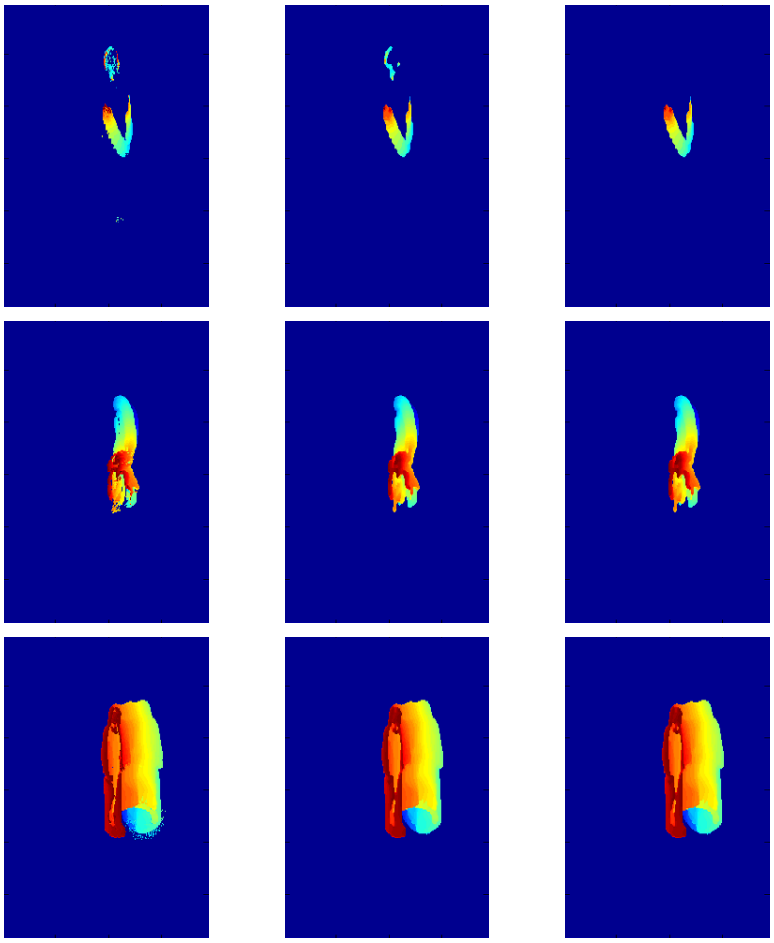
$$I \bullet B = (I \oplus B) \ominus B \tag{3.7}$$

Que es a su vez el resultado de realizar una operación de erosión a la imagen y seguidamente aplicar la operación de dilatación con el mismo elemento estructural B.

Después del filtrado morfológico se ha empleado el análisis de regiones sobre la MHI filtrada. Las capacidades del análisis de regiones permiten identificar y extraer características de regiones de píxeles conectados entre sí, o *blobs*, dentro de una imagen. Un blob es una región conectada por píxeles de forma que entre dos píxeles cualesquiera de dicha región se puede encontrar un camino conectando dichos píxeles. Mediante el concepto de conectividad se quiere expresar que dos píxeles pertenecen al mismo objeto. A través del estudio de la conectividad de los píxeles se analizan los blobs que conforman la MHI, eliminando aquellos con menor área. Esto eliminará regiones ruidosas debidas a sombras o movimientos residuales que no forman parte de la acción que se está ejecutando.

3.2. Dirección del movimiento

Por la forma de construcción de la MHI (*ecuación 3.1*) siguiendo la ejecución temporal de la acción, ésta es sensible a la dirección del movimiento. Por lo que a partir de estas plantillas es posible obtener información sobre la dirección de dicho movimiento directamente del gradiente



(a) **(b)** **(c)**

Figura 3.3: *(a) MHI de distintas acciones (mirar el reloj, sentarse y andar) por la ecuación 3.1. (b) MHI después del filtro morfológico. (c) MHI después del análisis de blobs.*

de intensidad en el interior de la imagen MHI. Los vectores de gradiente serán ortogonales a los bordes producidos por los distintos niveles de gris dentro de la imagen, los cuales se corresponden a un escalón temporal en la ejecución del movimiento. Este es un concepto similar al de flujo óptico.

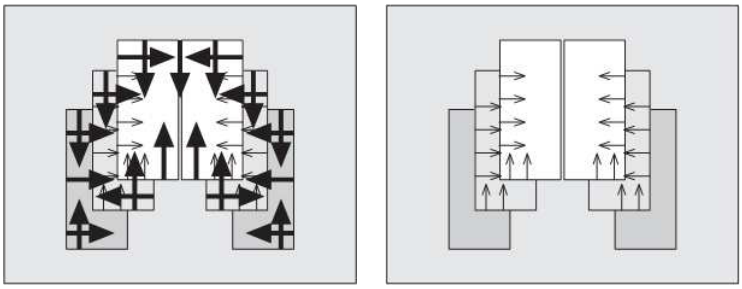
Estos gradientes se pueden calcular eficientemente mediante la convolución de la MHI con las máscaras del operador Sobel para calcular las derivadas parciales en dirección horizontal y vertical [4], como en las ecuaciones 3.8 y 3.9.

$$S_x(x,y) = MHI * \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{3.8}$$

$$S_y(x,y) = MHI * \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{3.9}$$

De estas derivadas parciales podemos calcular la orientación del gradiente en cada píxel como:

$$\theta(x,y) = \arctan\left(\frac{S_y(x,y)}{S_x(x,y)}\right) \tag{3.10}$$



(a) **(b)**

Figura 3.4: *(a) Vectores de gradiente de intensidad calculados sobre toda la imagen. (b) Vectores de gradiente válidos.*

Hay que tener cuidado al calcular los vectores de gradientes porque sólo son válidos en el interior de la región de movimiento en la MHI. Se descarta también el borde alrededor de esta región porque píxeles con valor cero (región sin movimiento) se incluyen en el cálculo de los gradientes produciendo orientaciones erróneas y de grandes magnitudes. Tampoco se tendrán en cuenta los píxeles que tengan un contraste demasiado alto entre ellos, lo que es debido a una larga diferencia temporal. Entonces, nos centraremos en saltos temporales entre fotogramas consecutivos, es decir, aquellos píxeles que contengan una diferencia máxima de intensidad igual a uno en un entorno 3x3 centrado en en ese píxel (*figura 3.4*).

Capítulo 4

Extracción de características

La selección de las características que representen la información más relevante de la secuencia de imágenes es una importante cuestión. Nosotros usamos un nuevo descriptor con información relativa al movimiento del sujeto basado en los vectores de gradiente de intensidad calculados en el capítulo anterior. De esta forma, la acción humana puede concebirse como una sucesión de pequeños movimientos. Para hacer más robusta esta información vamos a tomar la localización de la cabeza como referencia espacial dentro de la imagen a la hora de normalizar. Así se construirá un histograma bidimensional de gradientes según su orientación y su posición angular respecto a la cabeza. Como paso previo se hace necesario localizar la cabeza del sujeto dentro de la imagen. Con ese objetivo, en este proyecto se ha desarrollado un detector de cabeza basado en siluetas que se describe en las siguientes secciones.

4.1. Detección y seguimiento de la cabeza

La detección y el seguimiento de la cabeza es un elemento muy importante en varias áreas de visión por computador, como la realidad virtual y la interacción hombre-máquina, siendo ampliamente utilizado en diversas aplicaciones, incluyendo el recuento de gente y la vídeo vigilancia. Existen diversas técnicas de detección de cabezas dependiendo de las características de las imágenes y el objetivo perseguido. En este caso se ha optado por un detector basado en siluetas a fin de localizar la cabeza de los sujetos y tomarla como referencia de los parámetros característicos de cada acción.

4.1.1. Detector de cabeza

Las bases de datos con las que trabajamos aquí no incluyen ninguna acción en la que el sujeto invierta su postura, es decir, se coloque con la cabeza hacia abajo. Luego suponemos, tanto en este detector como en el sistema de seguimiento del apartado 4.1.2, que la cabeza siempre estará localizada en el punto más alto del sujeto y, por consiguiente, lo mismo ocurrirá con su silueta. El uso de las proyecciones horizontal y vertical de la silueta para la detección de la cabeza, aunque no es el único método, es comúnmente usado por su simplicidad en casos donde la cabeza siempre se sitúa en la parte superior del sujeto [5, 11, 12]. Estas proyecciones se calculan como la suma de las intensidades de los píxeles por filas o por columnas respectivamente.

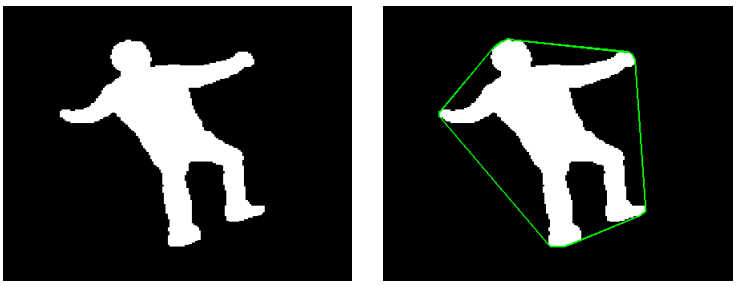


Figura 4.1: *Ejemplo de una silueta y su envoltura convexa.*

La información de las proyecciones no es suficiente en este caso porque en algunas acciones contenidas en la base de datos, por ejemplo la acción de rascarse la cabeza, los brazos estarían a la altura de la cabeza produciendo medidas erróneas en la proyección horizontal en esa región. Por eso, aquí se utiliza un sistema similar al que se usa en [12], que combina la forma geómetrica (*convex-hull* o envoltura convexa en el borde de la silueta) y la proyección horizontal de la silueta binaria para detectar la cabeza. Primero se localiza la coordenada vertical del punto más alto del sujeto como la primera fila distinta de cero en la proyección horizontal de la silueta. Después, se calcula la envoltura convexa, es decir, se calcula el polígono convexo más pequeño en el que esté contenida la silueta binaria del sujeto (*ver figura 4.1*). Por la topología del cuerpo, algunas partes del mismo siempre aparecerán en puntos de la envoltura convexa. Este es el caso de la cabeza, ya que es una forma convexa y, por tanto, estará presente en la envoltura convexa calculada. Buscamos estos puntos para encontrar posibles localizaciones de la cabeza en el borde de la silueta. Así pues, teniendo en cuenta la suposición hecha al principio, la cabeza se localiza en los puntos convexos situados a la altura de la coordenada vertical obtenida anteriormente. De hecho, suelen encontrarse dos puntos a la misma altura pero con distinta coordenada horizontal, que marcan los límites laterales de la parte más alta de la cabeza (*ver*

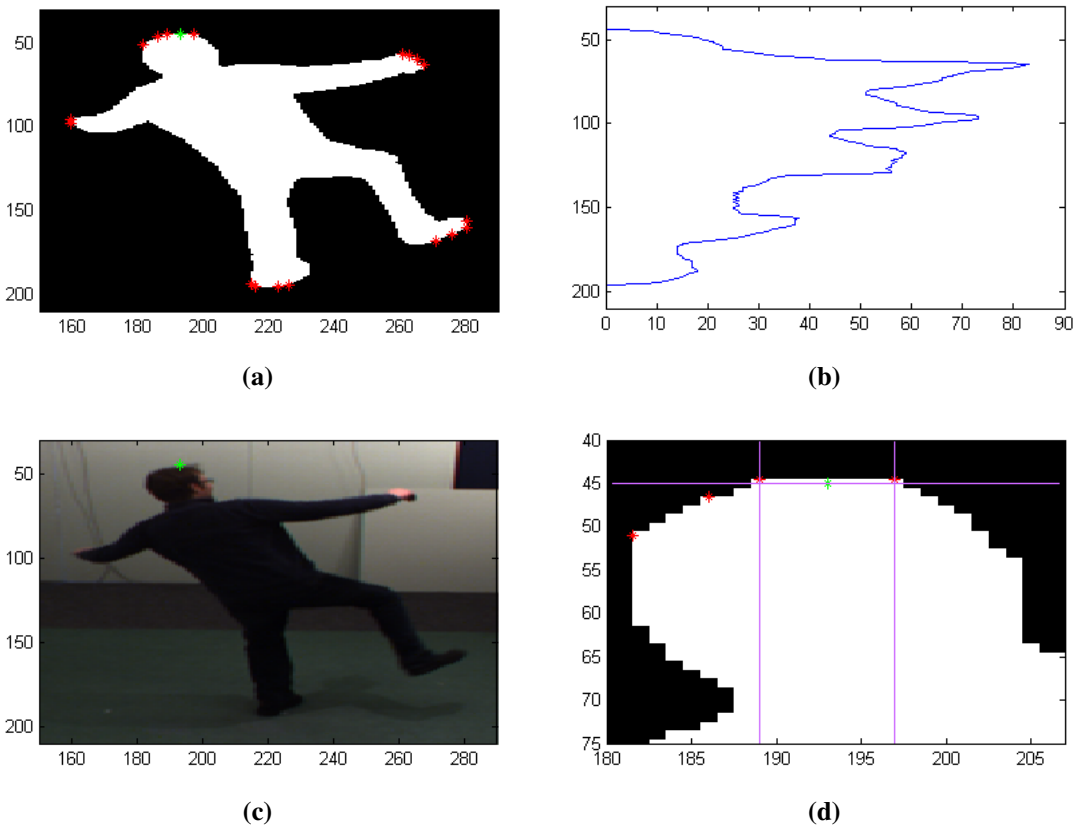


Figura 4.2: *(a) Silueta con los puntos de la envoltura convexa en rojo y punto de la cabeza en verde. (b) Proyección horizontal. (c) Fotograma original con el punto de la cabeza. (d) Detalle de la cabeza.*

figura 4.2d). En ese caso, la coordenada horizontal se calcula como el punto medio entre las coordenadas horizontales de los dos puntos convexos.

Este proceso se realiza para calcular el punto de la cabeza en cada uno de los fotogramas de las secuencias, quedándonos como referencia con el punto más alto. Estos datos se utilizarán más tarde en el paso de normalización de nuestro sistema como se explica en la sección 4.2 de esta memoria.

4.1.2. Seguimiento de la cabeza

Debido a la complejidad de las secuencias de la base de datos EPFL-IXMAS y los vacíos generados por las oclusiones en las siluetas obtenidas,se ha desarrollado un algoritmo de *tracking* o seguimiento de la cabeza para localizar el punto de referencia (*ver figura 4.3*). Hemos aprovechado el funcionamiento del detector de cabeza del apartado anterior por sus buenos re-

sultados y se ha añadido un sistema de seguimiento que delimita mucho la zona de búsqueda mediante el desplazamiento de una ventana. La idea es desplazar una ventana centrada en la cabeza según la información de movimiento procedente de la imagen MHI y realizar la búsqueda de la cabeza sólo en la parte de silueta dentro de la ventana.

Para empezar se necesita crear una ventana de las dimensiones de la cabeza y, para hacerla adaptable al sujeto, sus dimensiones serán función de la altura. A partir de la proyección horizontal de la silueta calculamos la altura y centramos la ventana en el punto alto de la cabeza según se refleja en la figura 4.3. Pero aún se necesita el punto alto de la cabeza para situar la ventana y comenzar el seguimiento. Por tanto, el primer paso del algoritmo de seguimiento será inicializar la posición de la cabeza y, para ello, el sistema toma el primer fotograma de la secuencia de vídeo y estima la posición de la cabeza mediante el detector del apartado 4.1.1 sobre la silueta completa de ese fotograma. La ventana de localización se centra en esa posición y comienza el seguimiento.

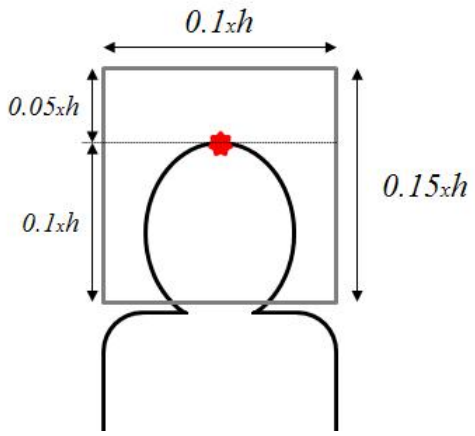


Figura 4.3: *Dimensiones de la ventana de búsqueda siendo h la altura del sujeto.*

Para desplazar la ventana se necesita saber dos cosas: hacia dónde se mueve y a qué velocidad. Esta información se obtiene a partir de las imágenes MHI. Como se ha descrito en la sección 3.1.2, se divide la secuencia de vídeo de cada acción en segmentos solapados y se calcula la imagen MHI para cada segmento. Esta segmentación se hace necesaria para evitar que los movimientos más recientes en la secuencia oculten aquellos que ocurrieron antes. Sobre estas imágenes se calculó el gradiente de intensidad y la orientación del movimiento como se explica en la sección 3.2. Como en este bloque sólo nos interesa el movimiento de la cabeza, para calcular la dirección del movimiento únicamente tendré en cuenta la región del interior de la ventana. A partir de estos vectores podemos calcular la orientación global del movimiento

como en [4] con la siguiente fórmula:

$$\overline{\theta} = \theta_{ref} + \frac{\sum_{x,y} (\theta(x,y) - \theta_{ref}) \times MHI_{norm}(x,y)}{\sum_{x,y} MHI_{norm}(x,y)} \tag{4.1}$$

En la ecuación 4.1, θ_{ref} es la referencia angular, calculada como el ángulo para el que se tiene el máximo del histograma de las orientaciones de todos los vectores de gradiente, $\theta(x,y)$. $MHI_{norm}(x,y)$ es la imagen MHI normalizada en el rango $[0,1]$ y las diferencias angulares, $(\theta(x,y) - \theta_{ref})$, son las mínimas diferencias angulares respecto a la referencia angular [1].

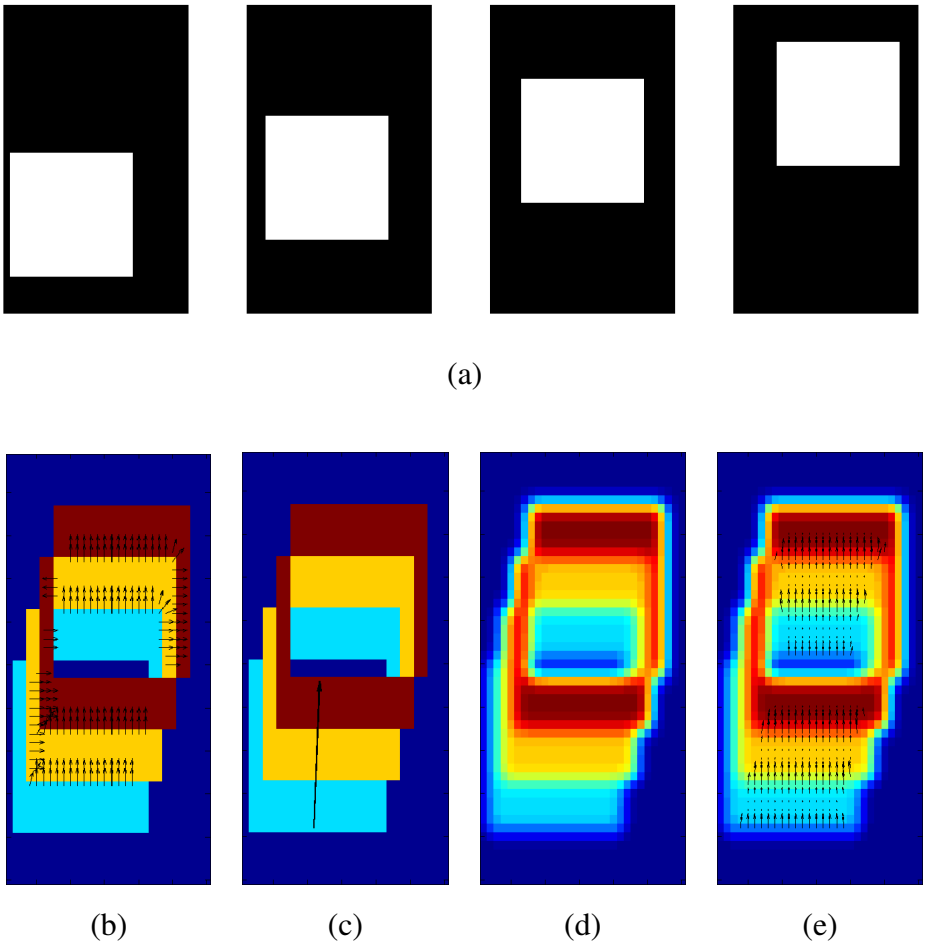
Para el cálculo de la velocidad se aplica un filtro gaussiano de suavizado a las MHI con el fin de reducir su sensibilidad al ruido antes de calcular los vectores de gradiente de intensidad sobre ellos. Después, una vez calculados estos vectores, se estima la velocidad media del movimiento medida en píxeles por fotograma como la inversa del módulo del vector promedio de los vectores del interior de la ventana y orientados en la dirección del movimiento:

$$v \simeq \frac{1}{\left| \frac{1}{K} \sum_{i=1}^{K} G_{\overline{\theta}}^{i} \right|} \tag{4.2}$$

siendo $\overline{\theta}$ la dirección del movimiento y $G_{\overline{\theta}}^{i}$ el vector i de gradiente orientado a $\overline{\theta} \pm 45^{\text{o}}$ que se encuentra dentro de la ventana. Se comprobaron resultados a partir de secuencias sintéticas generadas en el laboratorio con formas simples. Por ejemplo, en la figura 4.4 se puede ver el proceso realizado con un cuadrado que se desplaza 6 píxeles en vertical y 2 en horizontal, lo cual implica una velocidad de desplazamiento de 6.32 píxeles/fotograma; obteniéndose una estimación de 5.95 píxeles/fotograma.

Estimamos la posición de la cabeza en el siguiente fotograma desplazando la ventana en la dirección $\overline{\theta}$ tantos píxeles como nos indica el valor de velocidad. Buscaremos la cabeza con el detector del apartado 4.1.1 en la porción de silueta que queda dentro de la ventana. En caso de que la ventana se encontrase vacía porque no hay silueta en su interior, se incrementan las dimensiones de la ventana en un 50 %, sin desplazarla, en las siguientes iteraciones hasta que vuelve a encontrarse parte de la silueta en el interior de la ventana. Entonces el detector puede realizar la búsqueda en esa parte de silueta y la ventana vuelve a sus dimensiones iniciales. El siguiente paso, una vez localizada la cabeza, es volver a centrar la ventana en la posición alta de la cabeza antes de desplazarla y continuar la búsqueda en el siguiente fotograma (*figura 4.5*).

Este detector ofrece buenos resultados en el seguimiento, recuperándose de pérdidas y errores en la estimación de manera robusta. Pero todavía presenta algún problema cuando la silueta



(a)

(b) (c) (d) (e)

Figura 4.4: *Ejemplo de una secuencia sintética en la que se desplaza un cuadrado entre fotogramas (a). (b) MHI de la secuencia con los vectores de gradiente empleados en la estimación de la orientación del movimiento visible en (c). (d) MHI después del filtro gaussiano. (e) MHI con los vectores de gradiente que se usan en el cálculo de la velocidad media del movimiento.*

está dividida en varios blobs, como ocurre en los casos con oclusiones, donde parte del sujeto queda oculta en la escena. En estos casos, este seguimiento no es suficiente porque si la estimación fuera errónea y localizada fuera del blob donde realmente está la cabeza, puede darse el caso de que la ventana no se desplace lo suficiente como para volver al blob correcto. Por eso hemos introducido una incertidumbre en la búsqueda, de manera que además de buscar la cabeza en el interior de la ventana, la búsqueda se realiza también para toda la silueta. Si se encuentran dos estimaciones distintas de la cabeza y localizadas en blobs distintos, se incrementan las dimensiones de la ventana en un 50 % para la siguiente iteración y se continúa así hasta que ambas estimaciones coinciden entre sí o están en el mismo blob; entonces la ventana recupera las dimensiones iniciales.

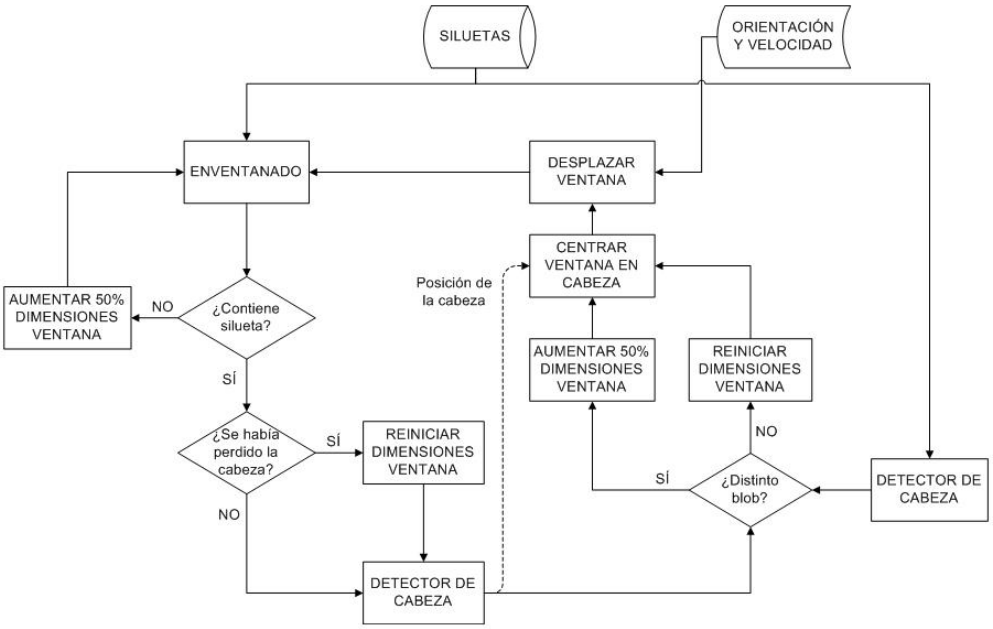


Figura 4.5: *Diagrama de flujo del algoritmo de seguimiento.*

4.2. Histogramas de gradientes

Una de las decisiones clave para el tipo de enfoque que se describe en este trabajo tiene que ver con la forma en la que se utiliza la información de las plantillas espacio-temporales extraídas de las secuencias de vídeo. La forma en que se representa esta información juega un papel fundamental. Dicha representación es, básicamente, un vector obtenido al procesar la información. Este vector será nuestro vector característico.

Como el objetivo es reconocer las actividades humanas de diferentes personas que se mueven libremente en el espacio, con diferentes orientaciones, así como con diferentes estilos y tamaños; es necesario algún proceso de normalización de los datos. Por ello, el objetivo es obtener descriptores que sean robustos a cambios en estas condiciones de captura y, a su vez, que constituyan representaciones compactas de la información obtenida de la MHI para reducir el coste computacional al trabajar con los mismos. Con esta idea hemos optado por localizar un punto de referencia estable, visible la mayor parte del tiempo y fácilmente predecible en la imagen: la cabeza del sujeto; tomando como referencia para la normalización su punto más alto

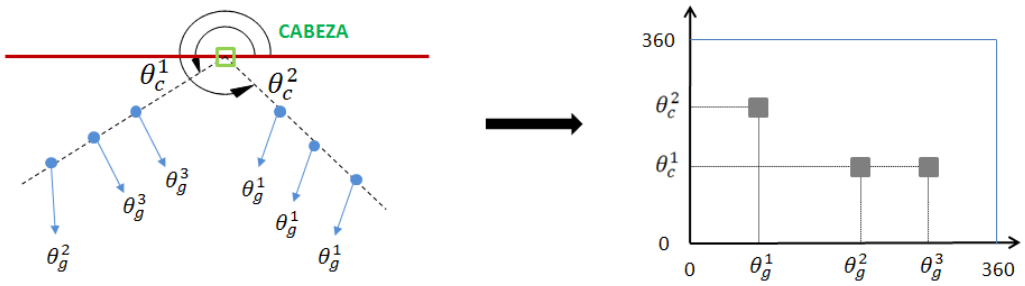
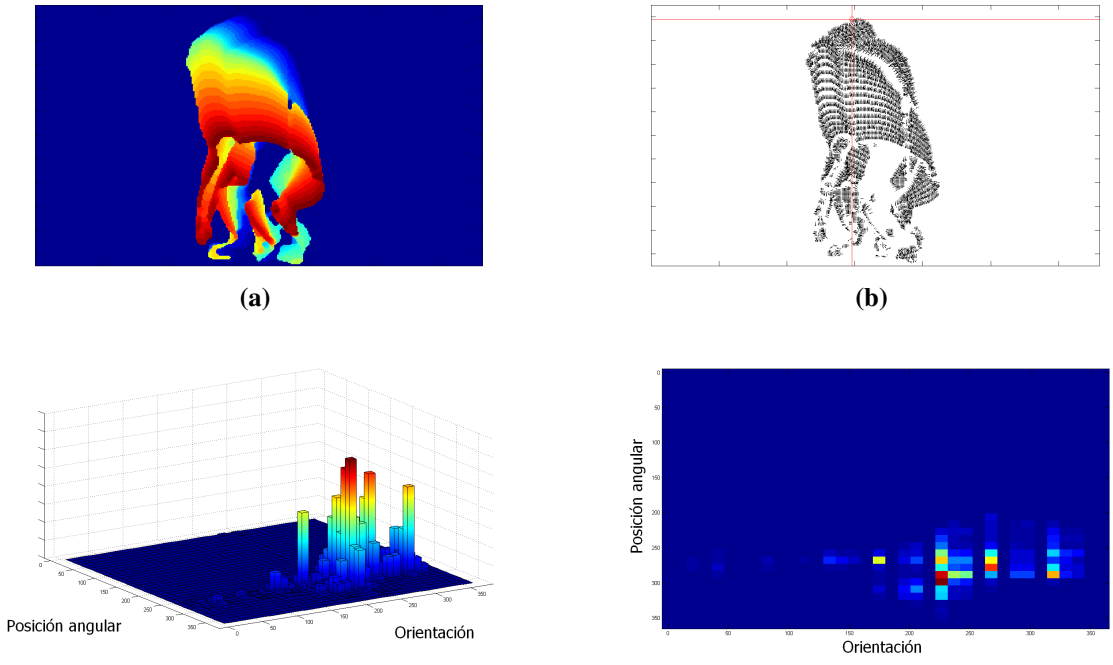


Figura 4.6: *Esquema de la construcción de los histogramas de gradientes a partir de los vectores de gradiente y la posición de la cabeza.*

dentro del segmento de vídeo del que procede la MHI. A partir de ahí se generan nuestros vectores de características, histogramas bidimensionales de gradientes de cada una de las plantillas MHI tomando como variables la orientación de los vectores de gradiente de intensidad (*sección 3.2*) y su posición angular respecto del punto alto de la cabeza (figura 4.6). Estos histogramas de gradientes o *mapas-2D* se dividen por su valor máximo, de forma que sus valores se encuentren en el rango $[0, 1]$.



(a) **(b)**

(c) **(d)**

Figura 4.7: *(a) MHI de un segmento de la acción "sentarse". (b) Vectores de gradiente de intensidad válidos y localización de la cabeza. (c) Histograma bidimensional de gradientes visto en 3D. (d) Histograma de gradientes en forma matricial.*

Estos histogramas de gradientes son matrices de tamaño 36x36 con bins de 10º, que nos generan un vector de características de 1296 componentes por cada segmento en que se divide la acción. En la figura 4.8 se muestran los promedios de los histogramas de gradientes para cada acción obtenidos al procesar las secuencias de la base de datos IXMAS.

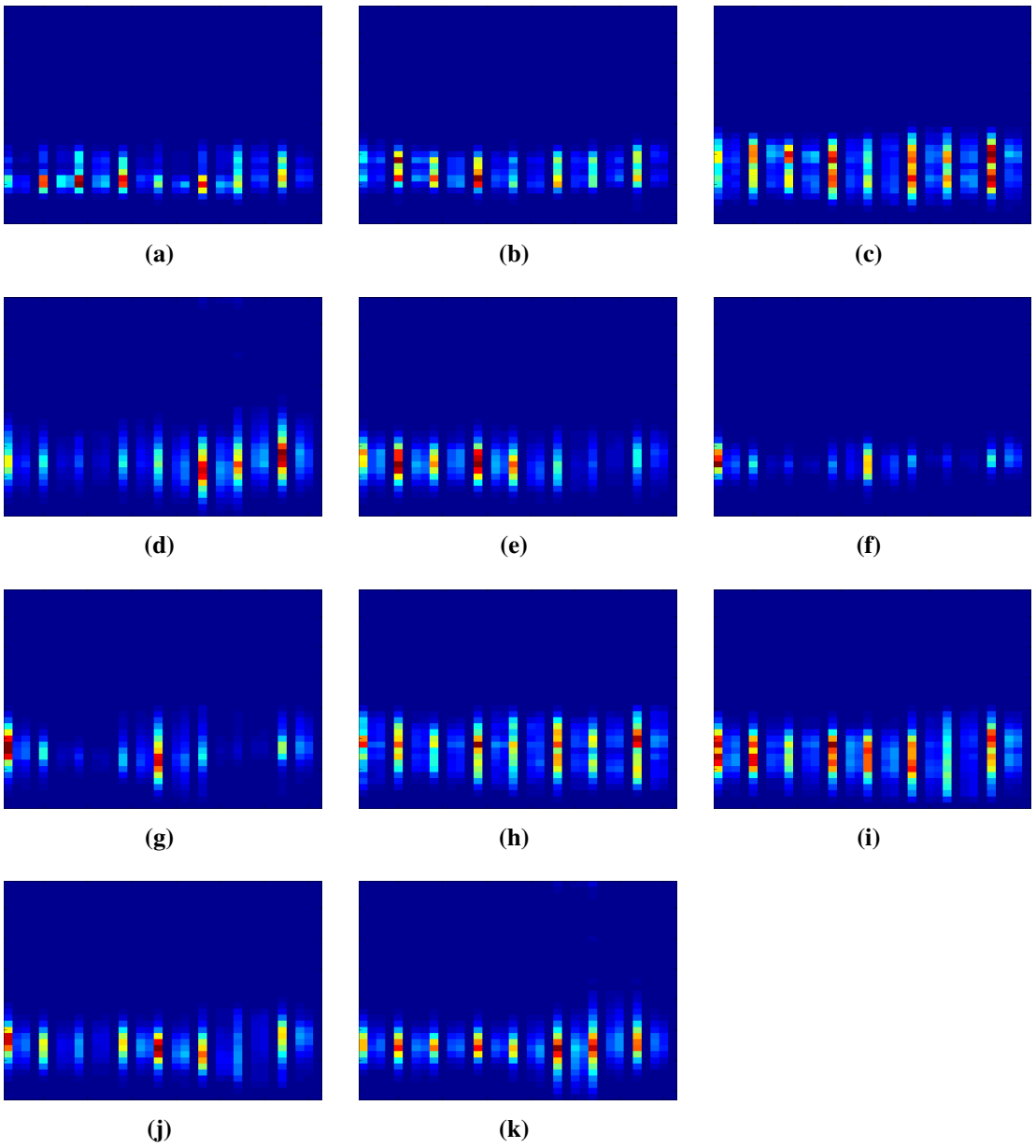


Figura 4.8: *Histogramas bidimensionales de gradientes obtenidos a partir de las secuencias de vídeo de la base de datos IXMAS para las acciones (a)-(k): mirar el reloj, cruzarse de brazos, rascarse la cabeza, sentarse en el suelo, levantarse, girar sobre sí mismo, caminar, saludar con la mano, dar un puñetazo, dar una patada y recoger algo del suelo.*

Capítulo 5

Diseño del clasificador

Una vez obtenido el vector de características que nos permita discernir las distintas acciones, es momento de completar el sistema de reconocimiento con la última de las etapas: la etapa de clasificación. Nuestro clasificador está basado en Modelos Ocultos de Markov (HMM - *Hidden Markov Models*) que serán los encargados de catalogar los vectores en cada una de las clases a las que pertenecen. Esta etapa consta de dos bloques bien diferenciados tanto por la matemática que en ellos se emplea como por el objetivo que se persigue con cada uno de ellos. Tenemos un bloque donde se reducen las dimensiones de los vectores característicos por medio del método PCA (*Principal Component Analysis*), seguido del bloque clasificador compuesto por modelos de Markov para cada una de las clases.

5.1. Reducción de la dimensionalidad

La reducción de la dimensionalidad suele ser útil en cualquier tarea de reconocimiento de patrones. Como se aprecia en la sección 4.2, el tamaño de nuestro vector de características es bastante grande y, por tanto, el número de parámetros que los modelos han de aprender. El aumento en la cantidad de dimensiones ocasiona un crecimiento exponencial del volumen del espacio en el que se encuentran los datos de entrenamiento. Este fenómeno, a su vez, provoca que los datos de entrenamiento se distribuyan de forma muy poco densa en el espacio, tornando difícil la tarea de encontrar patrones a partir de los mismos. Además, tanto a la hora de entrenar el clasificador como en el momento de clasificar un nuevo elemento, trabajar con vectores de menos dimensiones resulta beneficioso en cuanto a la potencia de cómputo necesaria. La herramienta ideal para resolver esta clase de problemas, reduciendo así el tamaño de este vec-

tor, es una de las técnicas más comunes para la compresión de la información, el análisis de componentes principales o PCA (*Principal Component Analysis*).

PCA es un método común de análisis de datos usado en reconocimiento de patrones, para eliminar la información redundante y reducir la carga computacional. Puesto que los patrones están ocultos en los datos pueden ser difíciles de encontrar cuando éstos tienen muchas dimensiones; así mediante el uso de esta técnica se pueden expresar los datos de modo que se destaquen sus similitudes y diferencias. La otra ventaja principal de PCA es que una vez encontrados los patrones en los datos se puede reducir su dimensionalidad sin tener gran pérdida de información. Así, el propósito de este método estadístico es condensar la información del amplio conjunto original de variables en otro conjunto con menos variables, que serán las componentes principales.

En este método se obtienen las componentes principales calculando los vectores propios de la matriz de covarianza de los datos, forzando previamente que estos tengan media cero. Estas componentes principales, como conjunto, forman una nueva base ortogonal para el espacio de los datos; es decir, son los ejes del nuevo sistema de coordenadas sobre los que se proyectan los datos. Después, las componentes principales se ordenan de forma decreciente según la información que contienen asociada a la varianza de las variables originales recogida en los valores propios de la matriz, λ. La primera componente principal será la que contenga la máxima varianza, la segunda componente la que contenga la segunda varianza más alta, y así sucesivamente. El conjunto de componentes principales construido de esta manera tiene las mismas dimensiones que el conjunto original de datos, pero, como suele ser habitual, nos quedamos con aquellas componentes principales cuya suma de varianzas contenga al menos el 95 % de la varianza total de los datos originales. Esta condición se expresa en la ecuación 5.1, donde λ_i es el valor propio de la *i*-ésima componente principal de las *N* componentes principales totales.

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \geq 0{,}95 \tag{5.1}$$

De esta forma conseguimos reducir nuestro vector de 1296 dimensiones a uno con 110 componentes realizando la transformación PCA sobre los *mapas-2D* generados a partir de la base de datos IXMAS. En la figura 5.1 se ve que la mayor parte de la variación en los datos puede ser representada por unos pocos vectores propios.

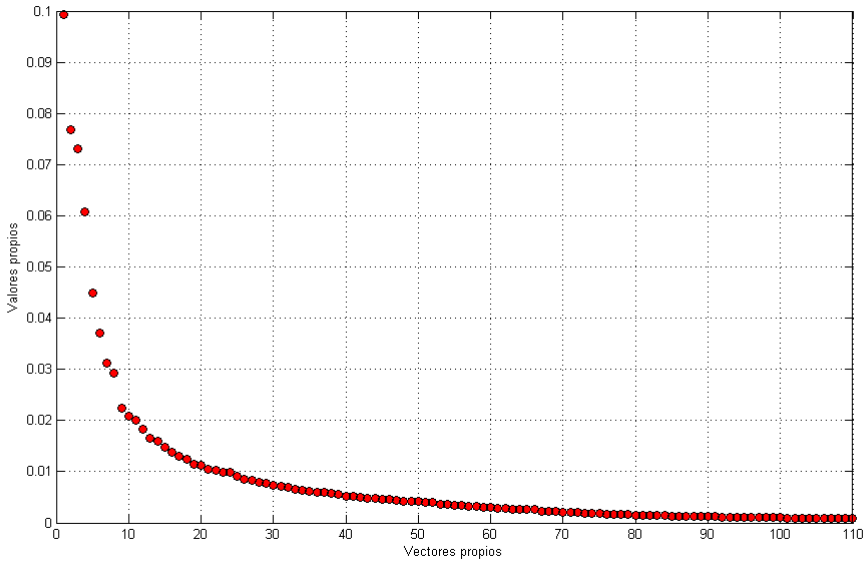


Figura 5.1: *Valores propios correspondientes a los 110 primeros vectores propios de la transformación PCA sobre todos los mapas-2D obtenidos para la base de datos IXMAS.*

5.2. Clasificador

5.2.1. Modelos Ocultos de Markov (HMM)

Un modelo oculto de Markov (HMM, del inglés *Hidden Markov Model*) es un autómata de estados finitos capaz de producir a su salida una secuencia de símbolos observable. Este autómata está formado por un conjunto de estados que se recorren a medida que el proceso evoluciona. Los cambios de estado se realizan según las probabilidades de transición de un estado a otro, mientras que se asocia una densidad de probabilidad a cada estado que define la probabilidad de emitir una observación cada vez que se produce una transición desde dicho estado del HMM. Por tanto, un HMM consta de dos procesos estocásticos: un proceso oculto que corresponde a la secuencia de las transiciones entre los estados, y un proceso observable que produce los símbolos.

Los elementos que constituyen un modelo de HMM son cinco [20]:

1) Un conjunto de N estados conectados entre sí de forma que cualquiera de ellos pueda ser alcanzado al menos desde un estado. Habiendo así diversos modos de interconexión, denotando como q_t al estado en que se encuentra el modelo en el instante t.

$$S = \{s_1, s_2, \ldots, s_N\}$$

2) Un conjunto de *M* símbolos observables que pueden ser generados en cada estado. Estos símbolos u observaciones corresponden a las salidas del sistema.

$$O = \{o_1, o_2, \ldots, o_M\}$$

3) La distribución de probabilidades de transición entre estados, definida como una matriz cuadrada de dimensión *N*, $A = \{a_{ij}\}$. Cada elemento a_{ij} corresponde a la probabilidad de transición del estado s_i al estado s_j, es decir, la probabilidad de estar en el estado *j* en el instante *t* si en el instante anterior se estaba en el estado *i*:

$$a_{ij} = P(q_t = s_j | q_{t-1} = s_i), \quad 1 \leq i, j \leq N$$

Por la naturaleza probabilística de los elementos del conjunto A, éstos deben cumplir:

$$0 \leq a_{ij} \leq 1, \quad 1 \leq i, j \leq N$$

$$\sum_{j=1}^{N} a_{ij} = 1$$

4) Las distribuciones de probabilidad de emisión de símbolos de salida para cada estado, $B = \{b_j(o_k)\}$, definidas como:

$$b_j = P(o_k | q_t = s_j), \quad 1 \leq j \leq N, \quad 1 \leq k \leq M$$

5) La distribución de probabilidades del estado inicial, $\Pi = \{\pi_i\}$, siendo π_i la probabilidad de que el estado inicial del HMM sea el estado s_i.

$$\pi_i = P(q_0 = s_i), \quad 1 \leq i \leq N$$

Como ocurría con las probabilidades de transición entre estados, los elementos del conjunto Π deben verificar:

$$0 \leq \pi_i \leq 1, \quad 1 \leq i \leq N$$

$$\sum_{i=1}^{N} \pi_i = 1$$

De esta forma, un HMM queda definido completamente al especificar los conjuntos Π, A y B que identifican al modelo λ como $\lambda = (\Pi, A, B)$.

5.2.2. Determinación de los parámetros del clasificador

Cuando los vectores característicos que se tratan son de naturaleza continua, como es nuestro caso, tratamos con HMMs continuos con una función de densidad de probabilidad de emisión aproximada por una mezcla de Gaussianas. Nuestro clasificador basado en HMM tendrá esta característica y estará formado por Q estados con observaciones en cada estado que estarán compuestas por la mezcla de M distribuciones Gaussianas. Este HMM tendrá una topología de izquierda-derecha cíclica similar a la que se representa en la figura 5.2. En esta sección obtendremos los valores de estos parámetros de diseño realizando el entrenamiento y la validación leave-one-subject-out variando el valor de M entre 1 y 10, y el número de estados, Q, entre 2 y el número de ventanas que utilizado para el análisis de cada acción y que será el que determina el máximo número posible de estados. Elegiremos posteriormente aquel caso que ofrezca el mejor porcentaje de reconocimiento en la validación.

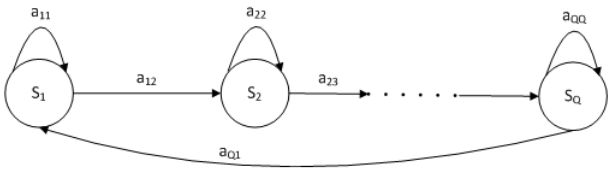


Figura 5.2: *HMM con topología izquierda-derecha cíclica como el usado en nuestro clasificador.*

El objetivo es entrenar un modelo por cada una de las acciones que hay que reconocer, por tanto, el entrenamiento de los modelos se realiza aisladamente para cada una de las clases o acciones. Este trabajo con Modelos Ocultos de Markov se lleva a cabo con el *Hidden Markov Model (HMM) Toolbox for Matlab* desarrollado por Kevin Murphy[1] que permite la implementación de los HMM en Matlab. Este software realiza un entrenamiento de los modelos siguiendo el algoritmo de Baum-Welch para la estimación de los parámetros del HMM (Anexo B). Este algoritmo es un caso especial del algoritmo *Expectation-Maximization* (EM), donde se realiza estimación del conjunto de parámetros iterativamente.

Los modelos se entrenan con los vectores procedentes de las secuencias de la base de datos IXMAS. Estas secuencias contienen las vistas de las 4 cámaras con lo que se pretende conseguir en la clasificación una independencia de la perspectiva que se tenga del sujeto. Para mejorar la validez estadística de los resultados se ha utiliza el método *leave-one-subject-out* de entrenamiento y validación. Empleando este método se entrena con las secuencias de 9 sujetos, es

[1] El Toolbox está disponible en http://www.cs.ubc.ca/ murphyk/Software/HMM/hmm.html

		Q			
		2	*3*	*4*	*5*
M	*1*	67,95	69,62	**73,33**	70,98
	2	65,30	67,20	63,56	64,47
	3	66,14	65,76	61,29	64,32
	4	67,35	66,52	64,85	63,33
	5	67,05	63,03	64,55	61,89
	6	65,30	64,62	65,53	63,94
	7	65,91	65,30	63,71	59,85
	8	64,85	65,83	63,86	62,12
	9	66,21	65,61	61,21	63,86
	10	65,98	63,56	65,08	58,71

Tabla 5.1: *Porcentajes promedio de reconocimiento resultantes del método leave-one-subject-out usando 5 ventanas para cada acción, donde se han variado el número de estados, Q, y el número de gaussianas, M. (En negrita el mejor caso).*

decir, con el 90 % de las secuencias; dejando un sujeto fuera, el cual se utilizará para el reconocimiento, lo que implica el uso de 108 secuencias de vídeo de cada acción para el entrenamiento de los modelos y 12 para el test. Esto se repite dejando fuera cada uno de los sujetos (12 secuencias) para un mismo valor de Q y M, promediando después los resultados de todos ellos. Como se menciona en la sección 3.1.2, se han realizado estas pruebas para 2 grupos de datos: los generados con 5 ventanas por cada acción y los generados con 10 ventanas.

En los resultados de reconocimiento obtenidos en la validación del clasificador con el 10 % de las secuencias siguiendo el método *leave-one-subject-out* (*tablas 5.1 y 5.2*) como se explica más arriba, se ve que los mejores porcentajes de reconocimiento se consiguen usando 5 ventanas en la segmentación de cada acción realizada. Además, el mejor valor se consigue para modelos con 4 estados ($Q = 4$) y distribuciones formadas por una sola Gaussiana ($M = 1$).

En la figura 5.3 se muestra la matriz de confusión promedio para el mejor caso. En cada una de las filas se presenta el porcentaje de reconocimiento para una misma acción; es decir, en la celda (p, q) de esta matriz se encuentra el porcentaje de casos en que la acción p ha sido clasificada en la clase q. Se aprecia que las acciones *mirar el reloj*, *cruzar los brazos*, *rascarse la cabeza* y *saludar con la mano* son susceptibles de confusión entre ellas aunque se tiene un reconocimiento bastante bueno para las dos primeras, con un 87.5 % y 72.5 % respectivamente. Esto es comprensible ya que en todas ellas el movimiento del brazo es similar en gran parte de

		Q								
		2	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
M	*1*	63,86	65,91	65,15	68,41	70,68	70,76	69,85	**70,98**	68,48
	2	61,59	62,73	61,67	60,15	60,00	59,70	58,71	59,47	57,65
	3	60,91	63,41	63,11	61,52	61,14	60,83	58,26	59,24	57,88
	4	61,74	61,21	60,30	60,91	61,21	60,45	56,97	54,55	57,42
	5	61,82	60,61	60,45	61,82	58,03	58,64	59,17	56,29	55,76
	6	59,77	60,68	61,06	58,26	56,82	55,00	60,08	55,68	51,74
	7	62,12	60,76	59,77	60,38	59,55	56,67	59,17	56,82	55,23
	8	59,39	62,95	57,05	60,76	57,20	57,73	52,88	57,50	50,83
	9	61,36	61,89	60,53	62,27	57,95	58,79	58,56	53,94	51,74
	10	61,97	62,65	60,53	58,41	60,08	55,76	51,82	52,95	51,29

Tabla 5.2: *Porcentajes promedio de reconocimiento resultantes del método leave-one-subject-out usando 10 ventanas para cada acción, donde se han variado el número de estados, Q, y el número de gaussianas, M. (En negrita el mejor caso).*

la realización. También se obtiene muy buenos porcentajes de clasificación para las acciones de *girar* y *caminar* como se podía intuir a la vista de los histogramas de gradientes de ambas acciones (*figura 4.8*).

En comparación con el estado del arte:

Método	ARR
Nevatia [17]	80.6 %
Cherla [6]	80.0 %
Liu [15]	78.5 %
Liu [16]	82.8 %
Yan [25]	78.0 %
Nuestro método	73.3 %

Tabla 5.3: *Porcentajes promedio de reconocimiento o Average Recognition Ratio (ARR) en comparación con otros métodos donde se trabaja con la base de datos IXMAS.*

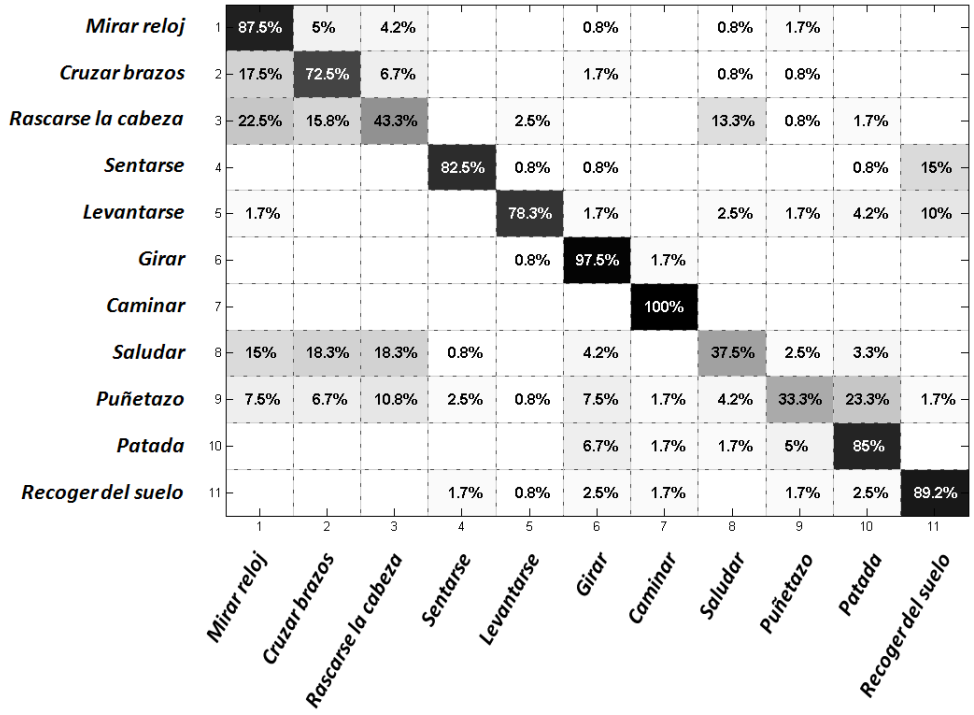


Figura 5.3: *Matriz de confusión promedio obtenida después de la validación leave-one-subject-out con secuencias de 5 ventanas para el clasificador con diseñado con $Q = 4$ y $M = 1$.*

No estamos lejos de estos valores teniendo en cuenta que nuestra propuesta incluye el detector automático de la cabeza y su seguimiento, factores importantes en la aplicación de este método en situaciones de oclusión parcial de la silueta; cosa que ninguno de los métodos comparados permite.

Capítulo 6

Análisis de resultados

En este capítulo analizaremos la respuesta del clasificador diseñado en el capítulo anterior. Para ello intentaremos clasificar secuencias de vídeo procedentes de dos bases de datos que contienen el mismo tipo de acciones pero en escenarios que pueden presentar oclusiones[1]. Estas bases de datos son: *Occluded IXMAS (O-IXMAS)* y *EPFL-IXMAS*. Como se recoge en el Anexo A de esta memoria, la base de datos O-IXMAS contiene las mimas secuencias que la base de datos IXMAS que se ha empleado en el entrenamiento y validación del clasificador (sección 5.2.2) a las que se les ha añadido una oclusión artificial; con la salvedad de que solamente se dispone de una ROI centrada en el sujeto y escala a una resolución de 48x64. Mientras la otra base, EPFL-IXMAS, consta de nuevas grabaciones de las mismas acciones recogidas en IXMAS en condiciones similares y desde distintos puntos de vista, donde cada acción se realiza 3 veces por cada sujeto: 2 secuencias en un escenario con oclusiones y 1 en un escenario libre de ellas. En la figura 6.1 se muestran algunas imágenes sacadas de las bases de datos con las que se trabaja en este proyecto: IXMAS en la fila superior, O-IXMAS en la fila central y EPFL-IXMAS en la fila de abajo.

Obviamente, se espera que en presencia de oclusiones la tasa de reconocimiento de nuestro sistema baje debido a la pérdida de información de los movimientos que quedan ocultos durante la ejecución de las acciones.

[1]La documentación de las bases de datos empleadas en este proyecto está disponible en el Anexo A

Figura 6.1: *Utilizamos distintas bases de datos en este proyecto. (Arriba) IXMAS, (Centro) O-IXMAS y (Abajo) EPFL-IXMAS.*

6.1. Base EPFL-IXMAS

Esta base de datos nos permite comprobar el funcionamiento de nuestro sistema tanto para casos con oclusiones como para situaciones similares a las empleadas en el entrenamiento del clasificador ya que contiene escenarios más realistas donde aparecen diversos objetos como mesas, sillas, etc. que ocultan parcialmente a los sujetos. Estos objetos se redistribuyen en el escenario para cada secuencia grabada, de forma que el escenario con oclusiones es distinto para cada una de ellas.

Disponemos en esta base de datos de nuevas secuencias de vídeo de las mismas 11 acciones que en IXMAS. Cada acción es realizada 3 veces por cada sujeto: 1 secuencia libre de oclusiones y 2 en un escenario con oclusiones. Estos vídeos se graban desde 4 cámaras que observan distintas perspectivas alrededor del actor. De nuevo, los actores eligen libremente su orientación, con lo que se espera lograr independencia en la clasificación respecto al punto de vista. En total se dispone de las secuencias de 6 sujetos para ser clasificadas (*ver anexo A*). Estas secuencias se dividen en 2 conjuntos según contengan o no oclusiones en el escenario, por tanto, trabajamos con 24 secuencias sin oclusiones y 48 secuencias con oclusiones para cada acción. En total son 792 secuencias entre los dos grupos.

Empleando para el reconocimiento las secuencias libres de oclusiones hemos logrado una tasa de acierto promedio del 69.70 %, un porcentaje ligeramente inferior al 73.33 % obtenido anteriormente. Si tenemos en cuenta que el escenario es diferente y, por tanto, las condiciones de adquisición de las imágenes; consideramos que el resultado obtenido es muy satisfactorio. En la figura 6.2 se presenta la matriz de confusión generada en esta clasificación. Ahí se puede

apreciar que las acciones mejor clasificadas siguen siendo *girar* y *caminar* con una clasificación correcta del 100 % y 95.8 % respectivamente. Sin embargo, las acciones *sentarse* y *recoger algo del suelo* se confunden más entre ellas de lo que reflejan los resultados obtenidos en la validación con la base de datos IXMAS (sección 5.2.2). Ambas acciones tienen parte del movimiento que es similar para ambas ya que el movimiento se realiza en la misma dirección y en la misma región respecto a la cabeza (270º aproximadamente).

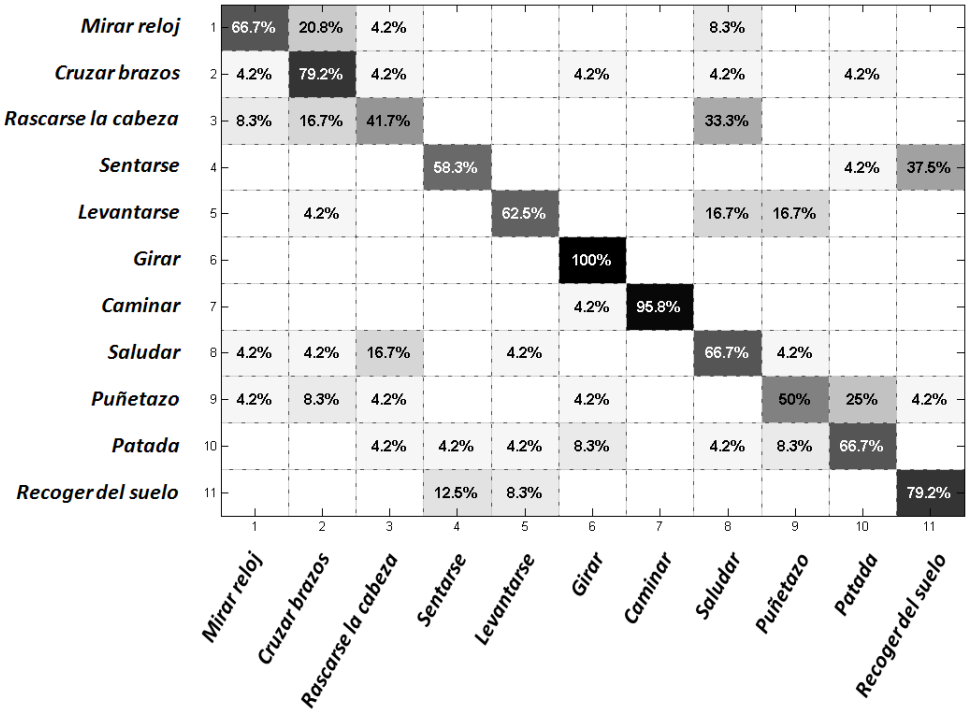


Figura 6.2: *Matriz de confusión promedio de la clasificación de las secuencias de vídeo sin oclusiones contenidas en la base de datos EPFL-IXMAS.*

Para el caso de secuencias con oclusiones obtenemos una clasificación correcta en un 63.26 % de las secuencias, generando la matriz de confusión de la figura 6.3. Como se esperaba, el porcentaje de reconocimiento es algo inferior al obtenido para secuencias sin oclusiones y, en general la confusión entre acciones se hace más notable en este experimento. La diagonal de la matriz se aleja del caso ideal del 100 % de reconocimiento incluso para la acción de *caminar*. Esta se confunde con la acción *girar* en algunos casos debido seguramente a la pérdida de información en algunos fotogramas por las oclusiones que ocultan el recorrido del sujeto en la escena. También se aprecia un aumento en la confusión entre aquellas acciones que consisten en movimientos de los brazos *saludar*, *rascarse la cabeza*, *mirar el reloj* etc.). Esta confusión puede entenderse por los movimientos tan similares que se ejecutan en casi todas ellas. Para

estos casos pensamos que se podrían conseguir mejores resultados modificando el número de ventanas en que se segmenta la acción, generando los mapas-2D a partir de secuencias de movimiento más cortas que recogiesen mejor el recorrido de los brazos en cada una de las diferentes acciones.

El caso de la acción *levantarse* es distinto, se puede ver en la matriz de confusión que el acierto a la hora de clasificar esta acción ha sido bastante pobre, sólo un 37.5 %. Además, la confusión con algunas de las acciones es notable. Estos resultados podrían ser debidos a una incorrecta localización de la cabeza en algunos momentos a causa de las oclusiones. Este punto se intentó mejorar con el trabajo con los blobs de la silueta en el seguimiento de la cabeza que se explica en la sección 4.1.2 de esta memoria, aunque a la vista de estos resultados sería conveniente revisar ese bloque.

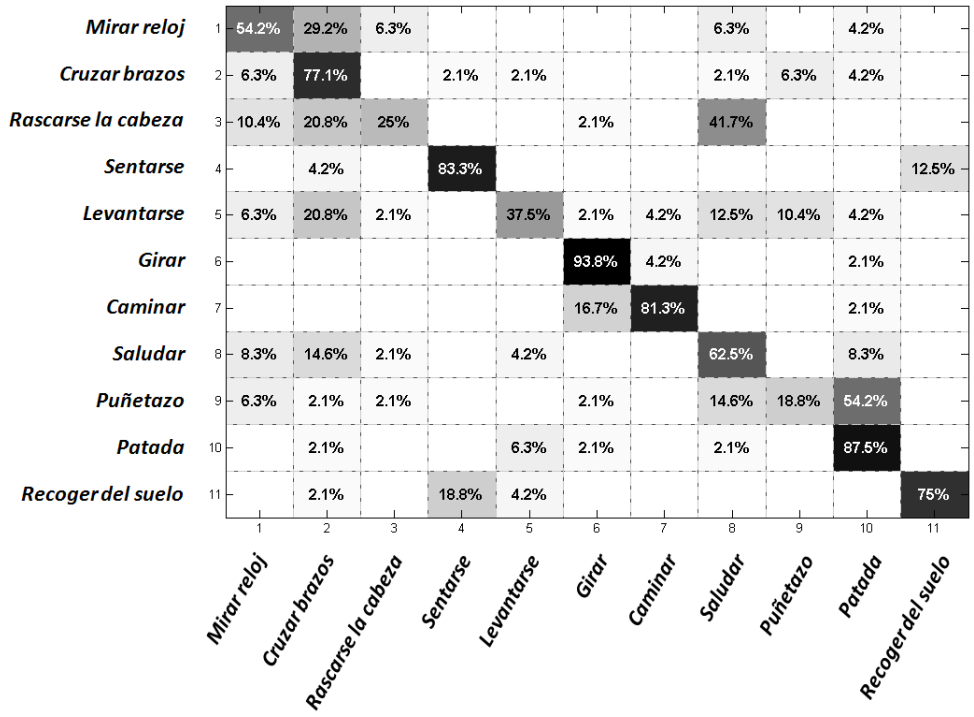


Figura 6.3: *Matriz de confusión promedio de la clasificación de las secuencias de vídeo con oclusiones contenidas en la base de datos EPFL-IXMAS.*

6.2. Oclusiones artificiales

En este apartado realizaremos el reconocimiento sobre las secuencias de vídeo contenidas en la base de datos O-IXMAS. Esta contiene las mismas secuencias que la base de datos IXMAS a las que se les han añadido oclusiones de forma artificial, superponiendo diferentes figuras sobre los fotogramas originales. Las nuevas secuencias constan de la región de interés (ROI) centrada en el sujeto y escaladas a un tamaño de 48x64. Las oclusiones artificiales son distintas para cada acción que realiza el sujeto y, también difieren para la misma acción realizada por distintas personas. Con esta base se ha conseguido una tasa de acierto global del 36.59 %, siendo

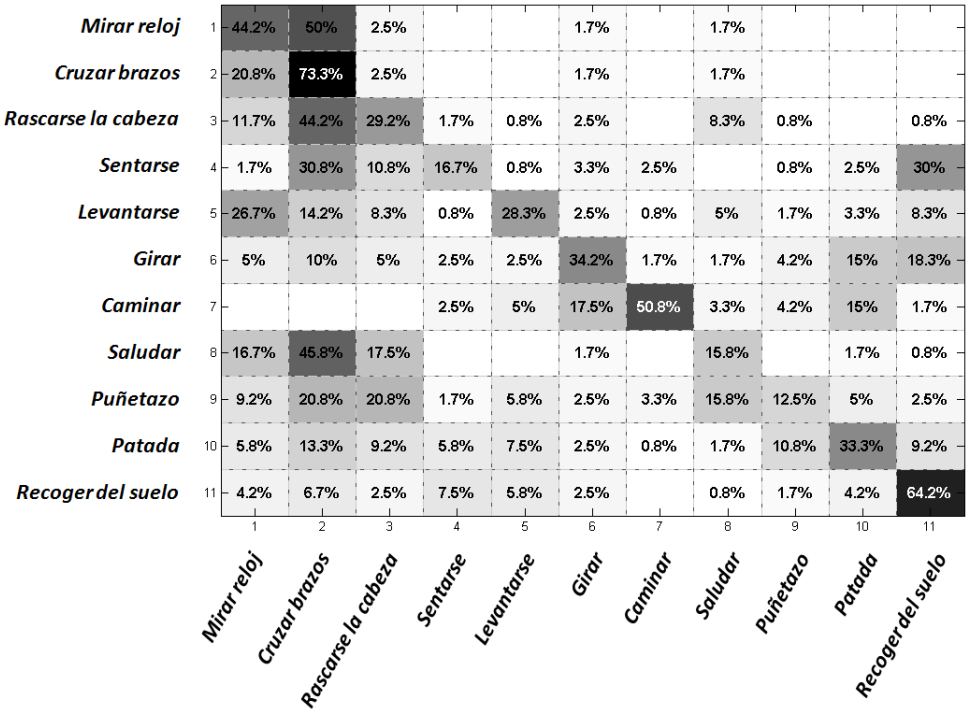


Figura 6.4: *Matriz de confusión promedio de la clasificación de las secuencias de vídeo sin oclusiones contenidas en la base de datos O-IXMAS.*

este resultado muy bajo incluso para secuencias con oclusiones. Se aleja considerablemente del 63.26 % obtenido para las secuencias con oclusiones de la base de datos EPFL-IXMAS. La matriz de confusión (*figura 6.4*) obtenida dista mucho de ser aceptable para un sistema de reconocimiento. Solamente 2 de las acciones superan el 60 % de clasificación correcta: *cruzar los brazos* y *recoger algo del suelo*.

Esta base de datos presenta problemas desde el principio debido a la escasa información que facilita y, en mayor grado, debido a que no se conoce el escalado exacto que se ha llevado a cabo respecto a la base de datos IXMAS. Este problema se presenta en la extracción de siluetas, que es la base de nuestra normalización, ya que la localización de la cabeza depende completamente de las siluetas de las que se disponga como se ve en la sección 4.1 de esta memoria. Además, las imágenes de que se dispone en esta base tienen una resolución muy baja lo que provoca tener pocos vectores de gradiente sobre la MHI para generar los histogramas bidimensionales de gradientes.

En la tabla 6.1 se hace una comparativa de los resultados obtenidos en este trabajo para todas las bases de datos empleadas.

	IXMAS	*O-IXMAS*	*EPFL-IXMAS sin oclusiones*	*EPFL-IXMAS con oclusiones*
ARR	73.33 %	36.59 %	69.70 %	63.26 %

Tabla 6.1: *Tabla resumen con los porcentajes promedio de reconocimiento (ARR) obtenidos usando nuestro método sobre estas bases de datos.*

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

Una vez finalizado el largo período de desarrollo de un curso docente de duración, llega el momento de comprobar hasta qué punto se han cumplido los objetivos propuestos.

Este proyecto me proponía el reto de estudiar la metodología existente en el campo del reconocimiento de acciones humanas, trabajando con herramientas desconocidas hasta este momento para mí. Por ello, partiendo como primer paso de la representación del movimiento utilizando la herramienta *Motion History Image*, el trabajo realizado ha sido enorme. De hecho, algunas partes del sistema pueden no ser óptimas por el tiempo requerido en su desarrollo. Este es el caso de la extracción de siluetas o el detector de cabeza, donde implementar algunos de los métodos existentes en la literatura hubiese exigido un tiempo añadido que habría sido excesivo sumado al que se ha necesitado para desarrollar nuestro sistema.

A pesar de ello, los resultados conseguidos han sido satisfactorios. Hemos podido presentar un nuevo descriptor para el reconocimiento de acciones a partir de los vectores de gradiente basándonos en la idea de la información que estos dan acerca de la dirección del movimiento. Este descriptor sumado a la segmentación de la acción en ventanas temporales ha conseguido tasas de reconocimiento muy buenas para determinadas acciones incluso en presencia de oclusiones, lo que nos hace pensar que se puede seguir investigando sobre esta herramienta para futuros trabajos.

El desarrollo completo de un sistema de reconocimiento como el aquí planteado es tremendamente ambicioso y requeriría varios proyectos más hasta conseguir un sistema sumamente eficiente, especialmente frente a casos donde se presentan oclusiones. Sin embargo, en mi opinión, este proyecto cumple las metas impuestas inicialmente; constituyendo el mismo la base en la que apoyarse para futuras investigaciones y trabajos en este campo.

7.2. Trabajo futuro

Para un futuro inmediato se podrían identificar varias líneas de acción con la intención de mejorar este sistema. Como primeras mejoras se podría actuar sobre los bloques que no han podido ser desarrollados eficientemente en este trabajo de forma que se eliminen fuentes de error en el sistema. Algunas sugerencias pueden ser:

- Eliminar la necesidad de las siluetas para el funcionamiento del sistema, puesto que son totalmente necesarias ahora mismo y el bloque de extracción de siluetas puede ser bastante complejo en algunas situaciones. Además, en este punto se suele introducir cierto error en la situación de la cabeza que se toma como referencia en la normalización, extendiéndose el error a lo largo del proceso de reconocimiento.
- Mejorar el bloque de detección y seguimiento de la cabeza con detectores más sofisticados como pueden ser los detectores basados en la forma de la cabeza, transformadas de Hough, histogramas de gradientes orientados (HOG), etc.

Incluso se podría pensar en ampliar el sistema incluyendo un bloque de forma que segmente automáticamente una secuencia de vídeo completa en la que se ejecutan diferentes acciones. Esto es un bloque que, dada una secuencia que contiene varias acciones, determine en qué instantes comienza y termina una acción. En este proyecto, estos instantes venían dados en la base de datos por lo que no ha sido necesario calcularlos.

Otra línea de acción que se propone es la de intentar optimizar el sistema modificando los distintos parámetros de diseño utilizados en el desarrollo del mismo:

- Optimizar el tamaño de las ventanas en que se segmentan las acciones para su posterior análisis. Se podría pensar en emplear ventanas de longitud variable según la duración de

las acciones o la velocidad de ejecución, que puede variar tanto entre sujetos como entre las distintas acciones.

- Procesar la MHI con algún filtro digital de suavizado como paso previo al cálculo de los gradientes de intensidad. De forma que los vectores de gradiente válidos no se encuentren solamente en saltos discretos de intensidad en el interior de la MHI.
- Modificar el tamaño de los bins usados en los histogramas bidimensionales de gradientes.
- Experimentar con mayor amplitud con bases de datos de acciones distintas a las utilizadas en este trabajo.

Anexo A

Bases de datos

En este anexo se va a profundizar más en las bases de datos utilizadas en el presente proyecto. Aquí van a exponer las principales características (contenido, finalidad, similitudes…), así como la forma en que son utilizadas para este trabajo. En total, se trabaja con tres bases de datos públicas multi-vista para el reconocimiento de acciones: *IXMAS*, *O-IXMAS* y *EPFL-IXMAS*; que se detallan en las secciones A.1, A.2 y A.3 respectivamente. La base de datos IXMAS es una base de datos multi-vista de referencia en la literatura del reconocimiento de acciones humanas y, las otras dos son una extensión o modificación de la primera. Estas bases de datos se basan principalmente en secuencias de vídeo grabadas desde varias cámaras estáticas en el escenario y en las que se realizan una serie de acciones cotidianas por actores no profesionales (*figura A.1*).

Figura A.1: *Utilizamos distintas bases de datos en este proyecto. (Arriba) IXMAS, (Centro) O-IXMAS y (Abajo) EPFL-IXMAS.*

A.1. Base de datos IXMAS

La base de datos IXMAS (Inria Xmas Motion Acquisition Sequences) es una base de datos multi-vista para el reconocimiento de acciones en la que se incluyen sujetos vistos desde puntos de vista arbitrarios [22, 24]. Aunque es comparable a otras bases de datos actuales para el reconocimiento de acciones se ha elegido por ser de uso público[1] y porque ya se había utilizado en otros trabajos realizados en este mismo laboratorio [19]. Se compone de 13 acciones cotidianas: *mirar el reloj*, *cruzarse de brazos*, *rascarse la cabeza*, *sentarse en el suelo*, *levantarse*, *girar sobre sí mismo*, *caminar*, *saludar con la mano*, *dar un puñetazo*, *dar una patada*, *apuntar con el dedo*, *recoger algo del suelo* y *lanzar un objeto*. Cada una de ellas llevada a cabo 3 veces por 12 personas (7 hombres y 5 mujeres). Un mismo sujeto realiza todas las acciones en una secuencia de vídeo, permaneciendo en reposo entre cada acción realizada. Las secuencias fueron grabadas con 5 cámaras calibradas y sincronizadas, proporcionando cada una de ellas un punto de vista diferente de la escena: cuatro cámaras alrededor del actor y una vista cenital (*ver figura A.2*). Para lograr la independencia de la vista, los actores son libres de cambiar su orientación en el escenario para cada secuencia grabada; aunque las cámaras permanecen estáticas en el escenario como se observa en la figura A.3. Todas estas secuencias fueron grabadas en un estudio con fondo estático.

Figura A.2: *Perspectivas vistas desde las 5 cámaras de la base de datos IXMAS para un mismo instante de la ejecución de una acción.*

Para ser consistente con trabajos anteriores realizados sobre esta esta base de datos se recomienda usar los datos de todas las cámaras excepto la vista cenital y de todos los sujetos excepto «*Pao*» y «*Srikumar*») [22, 24]. Además, solamente se realiza el estudio de las acciones {1 2 3 4 5 6 7 8 9 10 12} (todas menos *apuntar con el dedo* y *lanzar un objeto*. Luego, en total, trabajamos con los datos de 11 acciones realizadas 3 veces por 10 sujetos vistas desde 4 cámaras situadas alrededor suyo. Es decir, trabajamos con 330 secuencias multi-vista de acciones individuales o, si consideramos independiente cada punto de vista, son 1320 secuencias.

[1] La base de datos está disponible en la web del *INRIA's Perception Laboratory*: http://perception.inrialpes.fr, en la sección "Data".

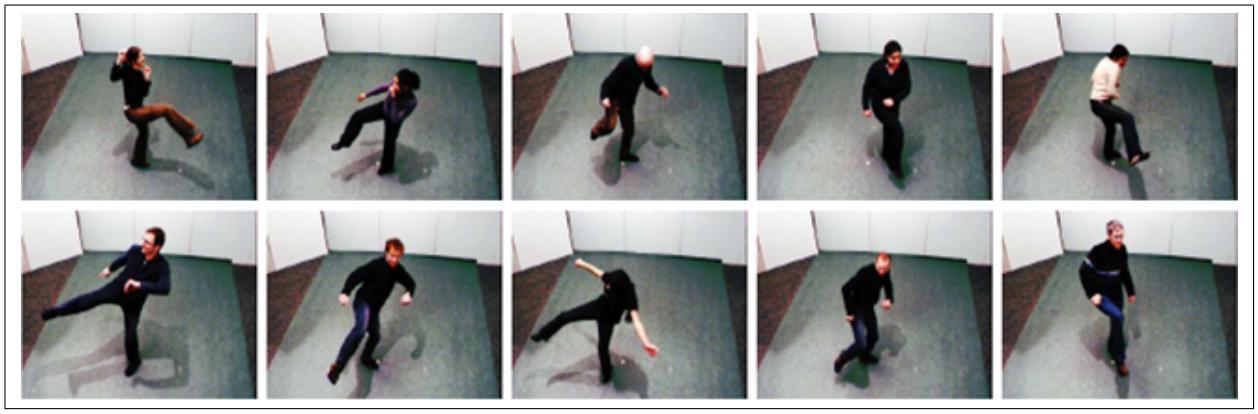


Figura A.3: *Misma perspectiva de la acción "dar una patada" realizada por varios sujetos. Aquí se ve claramente que los sujetos eligen libremente su orientación en el escenario.*

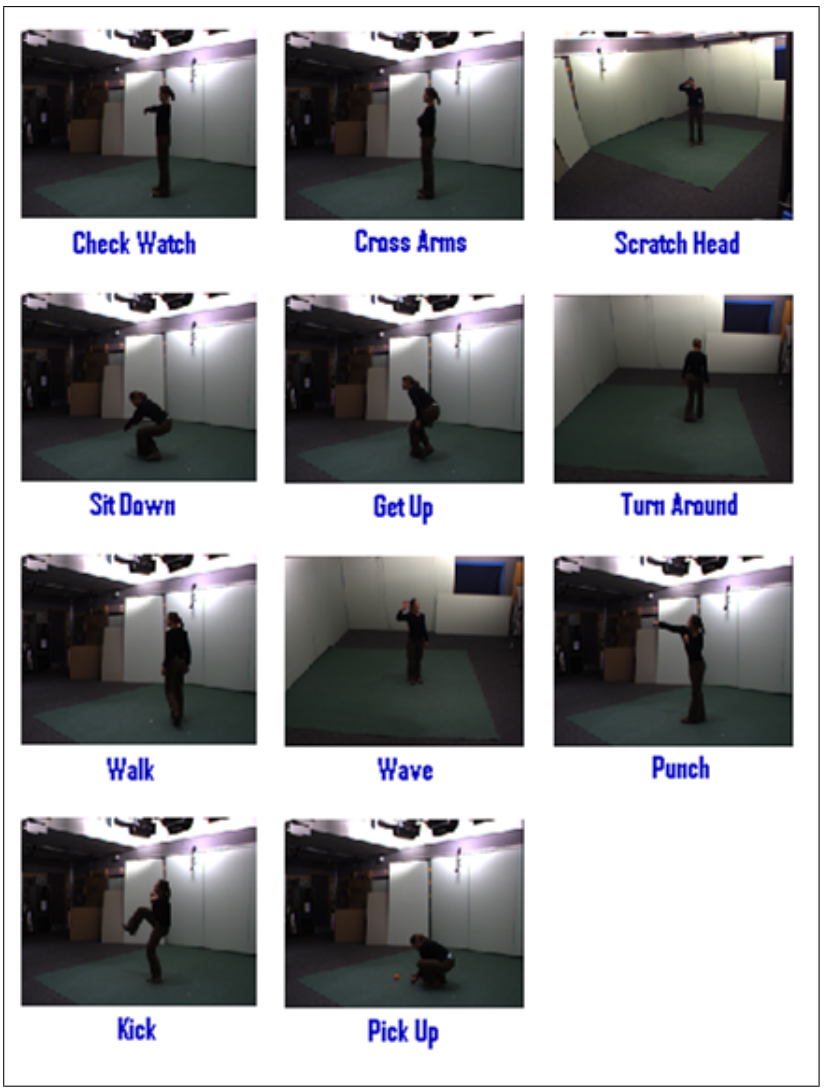


Figura A.4: *Distintas acciones sobre las que se realiza el estudio.*

Además de las secuencias de vídeo, esta base de datos dispone de más información que puede ser útil a la hora de procesar dichas secuencias. Todo el contenido de la base de datos IXMAS es el siguiente:

- Vistas originales de las 5 cámaras a 23 fps con una resolución de 390x291 en formato *PNG*.
- Siluetas binarias en formato *BMP* con una resolución de 390x291. Estas siluetas fueron obtenidas de los vídeos usando técnicas de sustracción de fondo [24].
- Volúmenes reconstruidos en formato Matlab de tamaño 64x64x64.
- *Ground Truth* a lo largo de los fotogramas. Esto nos indica la secuencia de fotogramas que se corresponden en el vídeo con cada acción realizada.
- Datos de calibración de las cámaras.

A.2. Base de datos O-IXMAS

Esta base de datos es una modificación de la detallada en el apartado A.1 aunque mucho menos completa ya que fue creada para un trabajo del laboratorio de visión por computador de la École Polytechnique Fédérale de Lausanne [23]. Esta modificación consiste en introducir oclusiones en los fotogramas de la base de datos IXMAS de forma artificial. Estas oclusiones están ubicadas de forma que aproximadamente la parte inferior del cuerpo, la parte derecha o la parte izquierda del sujeto queden ocultas (*figura A.5*). Estas oclusiones son distintas para cada acción que realiza el sujeto y, también difieren para la misma acción realizada por distintas personas. Para la notación seguida en este proyecto llamamos O-IXMAS (Occluded IXMAS) a esta base de datos.

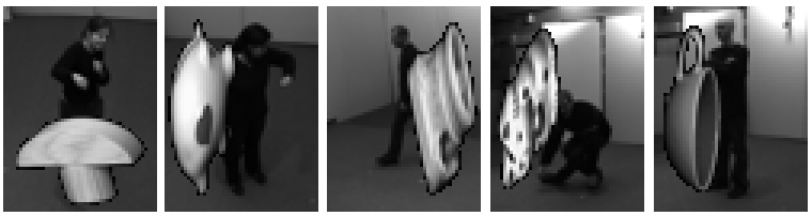


Figura A.5: *Imágenes de la base de datos O-IXMAS donde se aprecian las oclusiones artificiales.*

Puesto que es una modificación puntual, sólo se facilitan los datos en el formato que los autores han usado en sus trabajos. Esto es, una secuencia de vídeo para cada acción que realiza el sujeto. Estas secuencias se componen de fotogramas extraídos de la base de datos IXMAS, en los que se buscó una región de interés (ROI) centrada en el sujeto en el primer fotograma correspondiente a cada acción a partir de las siluetas de estos. Después estos fotogramas se normalizaron todos a una resolución de 48x64 para trabajar con ellos. En total se dispone de 1800 secuencias de vídeo, uno por cada acción realizada por los sujetos vistas desde cada una de las cuatro cámaras. El contenido que compone esta base de datos es el siguiente:

- Secuencias de vídeo en formato *AVI* con una resolución de 48x64.
- Archivos formato Matlab con la información de cada secuencia de vídeo.

A.3. Base de datos IXMAS con oclusiones

Esta base de datos fue desarrollada a semejanza de la base de datos IXMAS en el laboratorio de visión por computador de la École Polytechnique Fédérale de Lausanne [23]. Por ello en la notación de este proyecto nos referimos a esta base de datos como EPFL-IXMAS. Se compone de las mismas 11 acciones que IXMAS pero llevadas a cabo por diferentes actores; quienes además, en este caso, pueden estar parcialmente ocultos. Se pretende así manejar situaciones más realistas donde aparezcan oclusiones. Cada acción es realizada, en media, 3 veces y grabadas desde 5 cámaras que observan distintas perspectivas alrededor del actor. De nuevo, los actores eligen libremente su orientación. Las secuencias son grabadas en un entorno más realista con diversos objetos en el escenario que pueden ocultar al sujeto para algunos puntos de vista de la escena (*ver figura A.6*). Los objetos presentes en la escena se redistribuyen para cada secuencia grabada, así el escenario con oclusiones es distinto para cada secuencia. Las tres secuencias de cada sujeto de esta base de datos se dividen en dos conjuntos: secuencias sin oclusiones y secuencias con oclusiones.

Aunque esta base de datos es bastante completa, en algunos casos no se dispone de datos procedentes de todas las cámaras. Para las 2 secuencias de vídeo con oclusiones del actor «*Apu*» sólo hay imágenes de una de las cámaras; mientras que para una secuencia con oclusiones del actor «*Mustafa*» no hay datos de una cámara. Por esta razón, no entrarán en el estudio las secuencias de vídeo del actor «*Apu*» por falta de datos. Sin embargo, las del actor «*Mustafa*» sí que entrarán porque, como se comenta en el apartado A.1, en el estudio sobre la base de datos de

Figura A.6: *Perspectivas vistas desde las 5 cámaras de la base de datos EPFL-IXMAS para un mismo instante de la ejecución de una acción.*

acciones IXMAS sólo se trabajaba con las cámaras situadas alrededor del sujeto; por tanto, no se tendrán en cuenta las secuencias de la última cámara (cámara 4) para este proyecto. Aunque, en genenal, se dispone de 2 secuencias con oclusiones para cada sujeto, del actor «*Tola*» tenemos 3 secuencias, así que desechamos la última de estas para tener 2 secuencias con oclusiones de cada sujeto. Luego, en este proyecto se trabaja con 11 acciones realizadas 3 veces (1 secuencia sin oclusiones y 2 secuencias con oclusiones) por 6 sujetos y grabadas desde 4 cámaras situadas alrededor del sujeto. Se trabaja así con 264 secuencias de acciones individuales grabadas sin oclusiones en un escenario similar al de la base de datos IXMAS y, 528 secuencias de vídeo de acciones individuales que contienen objetos en el escenario ocultando parcialmente a los sujetos.

Esta base de datos, aunque no es tan completa como la base IXMAS, también proporciona más datos además de las secuencias de vídeo. Todo el contenido de la base de datos EPFL IXMAS es el siguiente:

- Vistas originales de las 5 cámaras a 23 fps en formato *AVI* con una resolución de 400x300.
- Vistas del background desde las 5 cámaras para dos situaciones distintas en el mismo formato que las secuencias anteriores.
- Datos de las *bounding box* o ROI que se toman centradas en el sujeto.
- *Ground Truth* a lo largo de los fotogramas. Esto nos indica la secuencia de fotogramas que se corresponden en el vídeo con cada acción realizada.
- Datos de calibración de las cámaras.

Anexo B

Modelos ocultos de Markov

Un modelo oculto de Markov (HMM, del inglés *Hidden Markov Model*) es un autómata de estados finitos capaz de producir a su salida una secuencia de símbolos observable [20]. Este autómata está formado por un conjunto de estados que se recorren a medida que el proceso evoluciona. Los cambios de estado se realizan según las probabilidades de transición de un estado a otro, mientras que se asocia una densidad de probabilidad a cada estado que define la probabilidad de emitir una observación cada vez que se produce una transición desde dicho estado del HMM. Por tanto, un HMM consta de dos procesos estocásticos: un proceso oculto que corresponde a la secuencia de las transiciones entre los estados, y un proceso observable que produce los símbolos. La figura B.1 representa un HMM de tres estados.

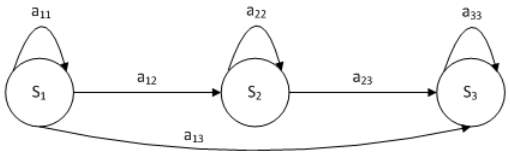


Figura B.1: *Ejemplo de un HMM de tres estados.*

B.1. Elementos de un modelo oculto de Markov

Los elementos que constituyen un modelo de HMM son cinco [20]:

1) Un conjunto de N estados conectados entre sí de forma que cualquiera de ellos pueda

ser alcanzado al menos desde un estado. Habiendo así diversos modos de interconexión, denotando como q_t al estado en que se encuentra el modelo en el instante *t*.

$$S = \{s_1, s_2, \ldots, s_N\}$$

2) Un conjunto de *M* símbolos observables que pueden ser generados en cada estado. Estos símbolos u observaciones corresponden a las salidas del sistema.

$$O = \{o_1, o_2, \ldots, o_M\}$$

3) La distribución de probabilidades de transición entre estados, definida como una matriz cuadrada de dimensión *N*, $A = \{a_{ij}\}$. Cada elemento a_{ij} corresponde a la probabilidad de transición del estado s_i al estado s_j, es decir, la probabilidad de estar en el estado *j* en el instante *t* si en el instante anterior se estaba en el estado *i*:

$$a_{ij} = P(q_t = s_j | q_{t-1} = s_i), \quad 1 \leq i, j \leq N$$

Por la naturaleza probabilística de los elementos del conjunto A, éstos deben cumplir:

$$0 \leq a_{ij} \leq 1, \quad 1 \leq i, j \leq N$$

$$\sum_{j=1}^{N} a_{ij} = 1$$

4) Las distribuciones de probabilidad de emisión de símbolos de salida para cada estado, $B = \{b_j(o_k)\}$, definidas como:

$$b_j = P(o_k | q_t = s_j), \quad 1 \leq j \leq N, \quad 1 \leq k \leq M$$

5) La distribución de probabilidades del estado inicial, $\Pi = \{\pi_i\}$, siendo π_i la probabilidad de que el estado inicial del HMM sea el estado s_i.

$$\pi_i = P(q_0 = s_i), \quad 1 \leq i \leq N$$

Como ocurría con las probabilidades de transición entre estados, los elementos del conjunto Π deben verificar:

$$0 \leq \pi_i \leq 1, \quad 1 \leq i \leq N$$

$$\sum_{i=1}^{N} \pi_i = 1$$

De esta forma, un HMM queda definido completamente al especificar los conjuntos Π, A y B que identifican al modelo λ como $\lambda = (\Pi, A, B)$.

B.2. Topologías

La topología de un HMM viene dada por el número de estados ocultos que lo componen, el número de símbolos y las transiciones de estados y emisiones de símbolos no permitidas (para las que las correspondientes probabilidades de transición o emisión se asume que son cero). Aunque hay muchas combinaciones posibles, los tipos más importantes de HMM son los de izquierda-derecha, o de Bakis, y los ergódicos. Estos últimos son el caso más genérico y se caracterizan porque cada estado del modelo puede ser alcanzado desde cualquier otro en un número finito de transiciones (figura B.2). Los HMM con arquitectura de izquierda-derecha tienen la propiedad de que el índice de los estados hacia los que el proceso evoluciona con el tiempo es creciente ($a_{ij} = 0$, $j < i$), es decir, el sistema progresa de izquierda a derecha (figura B.1). Este hecho hace que la matriz de transiciones, A, sea triangular superior para un HMM con esta topología; es decir, si en un momento determinado el modelo está en el estado s_i, seguirá en dicho estado con probabilidad a_{ii} o bien evolucionará a un estado s_j, con $j > i$, con probabilidad a_{ij}.

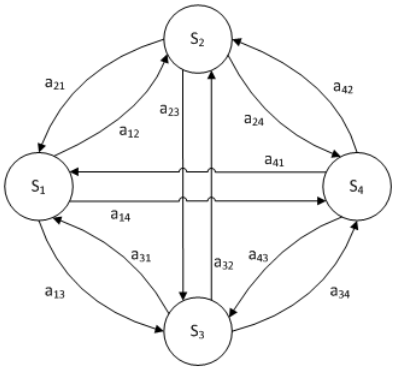


Figura B.2: *Ejemplo de un HMM ergódico de cuatro estados.*

B.3. Los tres problemas básicos del modelado HMM

Una vez definido el modelo del HMM surgen tres problemas básicos de interés que deben resolverse de cara a posibles aplicaciones prácticas en el reconocimiento:

1) **Evaluación.** Dada una secuencia de observaciones $X^\tau = x_1 x_2 \cdots x_\tau$ y un modelo del

HMM λ, se busca la probabilidad de que la secuencia observada haya sido producida por dicho modelo, $P(O|\lambda)$.

2) **Decodificación.** Dada una secuencia de observaciones $X^\tau = x_1 x_2 \cdots x_\tau$ y un modelo λ, se requiere obtener la secuencia de estados $Q^\tau = q_1, q_2, \ldots, q_\tau$ óptima que haya podido producir la secuencia observada.

3) **Apredizaje.** Dada una secuencia de observaciones $X^\tau = x_1 x_2 \cdots x_\tau$ y un modelo λ, se necesita estimar los parámetros del modelo $\lambda = (\Pi, A, B)$ de forma que se maximice la probabilidad de generación de dicha secuencia por el modelo, $P(O|\lambda)$.

Para un sistema de reconocimiento basado en HMM estos problemas se concretan en las fases de entrenamiento y reconocimiento. En la fase de entrenamiento del modelo surge el problema de aprendizaje; a un modelo se le proporciona una secuencia de observaciones para estimar los parámetros A, B y Π. Mientras, la solución al problema de evaluación permitirá evaluar la probabilidad de generación de una secuencia de observaciones por el modelo, lo que puede utilizarse para clasificar secuencias de observaciones. El problema de decodificación, una vez solucionado, permitirá extraer información sobre el proceso oculto; esta información puede ser útil a la hora de interpretar el significado de los estados del modelo.

B.3.1. Evaluación de secuencias

Los HMM se pueden utilizar para evaluar la probabilidad de que una secuencia haya sido producida por un modelo basándose en el proceso de producción de secuencias, $P(O|\lambda)$. Un algoritmo eficiente para evaluar la probabilidad anterior es el proceso *Forward-Backward*. Las probabilidades hacia delante o *forward*, $\alpha_i(t)$, se definen como la probabilidad de obtener la secuencia parcial de observaciones hasta el instante t estando en el estado s_i en dicho instante para el modelo λ:

$$\alpha_i(t) \equiv P(x_1 x_2 \cdots x_t, q_t = s_i|\lambda) \tag{B.1}$$

Es posible calcular $\alpha_i(t)$ de manera recursiva a partir de la inicialización para $t = 1$.

$$\alpha_i(1) = P(x_1, q_t = s_i|\lambda) = \pi_i b_i(x_1) \tag{B.2}$$

Para los instantes sucesivos se puede expresar en función de los valores anteriores teniendo en cuenta las probabilidades de transición y de producción de símbolos.

$$\alpha_i(t) = \sum_{j=1}^{N} \alpha_j(t-1) \cdot a_{ji} \cdot b_i(x_t), \quad 1 < t \leq T, \quad 1 \leq i \leq N \tag{B.3}$$

Finalmente, la probabilidad total de obtener la secuencia observada será la suma de las correspondientes a cada estado considerado como final.

$$P(X^T|\lambda) = \sum_{i=1}^{N} \alpha_i(T) \tag{B.4}$$

Las probabilidades hacia atrás o *backward*, $\beta_i(t)$ se definen como la probabilidad de la secuencia parcial constituida por todos los símbolos presentados a partir del instante t cuando el estado en dicho instante es s_i:

$$\beta_i(t) \equiv P(x_{t+1}x_{t+2}\cdots x_T, q_t = s_i|\lambda) \tag{B.5}$$

De las misma forma que para las probabilidades hacia delante, es posible calcular $\beta_i(t)$ de manera recursiva si se inicializa a 1 para $t = T$.

$$\beta_i(t) = \sum_{j=1}^{N} \beta_j(t+1) \cdot a_{ij} \cdot b_j(x_{t+1}), \quad 1 < t \leq T-1, \quad 1 \leq i \leq N \tag{B.6}$$

B.3.2. Decodificación

La decodificación de la secuencia de estados más probable, dada una secuencia X^T y un modelo λ, no tiene una única solución pues ésta depende del criterio utilizado para determinar la mejor secuencia. El criterio que se suele utilizar para escoger la secuencia óptima de estados Q^{T^*} es maximizar la probabilidad condicionada de generación de la observación.

$$Q^{T^*} = \arg\max_{Q^T} P(X^T|Q^T, \lambda) \tag{B.7}$$

La resolución de esta ecuación se hace mediante el algoritmo de Viterbi [20]. Se trata de un algoritmo iterativo en el que se define una variable $\delta_i(t)$ como la probabilidad máxima de generación de una secuencia de t símbolos sobre cualquier secuencia simiple de estados cuyo estado final es el s_i.

$$\delta_i(t) \equiv P^*(x_1x_2\cdots x_t|q_t = s_i, \lambda) \tag{B.8}$$

Por inducción se verifica la siguiente igualdad:

$$\delta_i(t) = \max_{1 \leq j \leq N} \{\delta_j(t-1)a_{ji}\}b_i(x_t) \tag{B.9}$$

Para conocer la secuencia de estados es necesario almacenar los valores del argumento que maximizan $\delta_i(t)$. Para ello se utiliza una matriz en la que cada elemento, $\Psi_{it} \equiv \Psi_i(t)$, contiene el índice del estado que maximiza la expresión anterior en el instante t. A partir de aquí han de establecerse la condición inicial, la condición de parada y cómo realizar las iteraciones.

B.3.3. Entrenamiento de modelos

El problema del entrenamiento o aprendizaje del modelo, como se ha explicado anteriormente, implica la estimación de los parámetros del modelo $\lambda = (\Pi, A, B)$, dada la secuencia de observaciones de entrenamiento $X^T = x_1 x_2 \cdots x_T$, de manera que se maximice la probabilidad $P(X_T|\lambda)$. Este proceso presenta grandes dificultades, pues no se conoce la solución analítica del problema [20]. Generalmente, se adopta una solución basada en la reestimación iterativa de los parámetros del modelo hasta alcanzar un punto óptimo local de dicha probabilidad. Este método iterativo suele ser el algoritmo de Baum-Welch, un caso particular de algoritmo EM (*Expectation-Maximization*). Un algoritmo EM permite encontrar estimadores de máxima verosimilitud de los parámetros en modelos probabilísticos, aun cuando en el conjunto de entrenamiento falten datos, como en el caso de los HMM que faltarían los estados que se encuentran ocultos. El funcionamiento del procedimiento iterativo parte de un modelo inicial que se puede seleccionar aleatoriamente y se realiza el cálculo de las transiciones y símbolos emitidos que son más probables según el modelo escogido. A partir de estos cálculos, se construye un nuevo modelo, $\lambda' = (\Pi', A', B')$, en el que se incrementa la probabilidad de transición y generación de símbolos determinadas en el paso anterior. Así, para la secuencia de observaciones de entrenamiento $P(X^T|\lambda') > P(X^T|\lambda)$, es decir, es más probable que la secuencia de observaciones de entrenamiento haya sido generada por el modelo estimado que por el modelo anterior. Este proceso se repite hasta que se satisface algún criterio de parada: la diferencia en las probabilidades de un modelo a otro es imperceptible (se alcanza la convergencia del método) o se alcanza el número máximo de iteraciones fijado. De esta forma, el algoritmo Baum-Welc garantiza la convergencia uniforme hacia un máximo local de la función de probabilidad de generación.

Bibliografía

[1] *Open Source Computer Vision Library Reference Manual*. Intel Corporation, 2001.

[2] AHAD, M., TAN, J., KIM, H., AND ISHIKAWA, S. Motion history image: Its variants and applications. *Machine Vision and Applications 23*, 2 (2012), 1–27.

[3] BOBICK, A. F., AND DAVIS, J. W. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 3 (2001), 257–267.

[4] BRADSKI, G. R., AND DAVIS, J. W. Motion segmentation and pose recognition with motion history gradients. *International Journal of Machine Vision and Applications 13*, 3 (2002), 174–184.

[5] CERLINCA, T. I., PENTIUC, S. G., VATAVU, R. D., AND CERLINCA, M. C. Hand posture recognition for human-robot interaction. In *Proceedings of the 2007 Workshop on Multimodal Interfaces in Semantic Interaction* (2007), pp. 47–50.

[6] CHERLA, S., KULKARNI, K., KALE, A., AND RAMASUBRAMANIAN, V. Towards fast, view-invariant human action recognition. In *IEEE Computer Vision and Pattern Recognition for Human Communicative Behaviour Analysis CVPR4FB'08* (2008).

[7] DAVIS, J. W. Recognizing movement using motion histograms. Technical Report 487, MIT Media Lab., 1999.

[8] DAVIS, J. W. Hierarchical motion history images for recognizing human motion. In *IEEE Workshop on Detection and Recognition of Events in Video* (2001), pp. 39–46.

[9] FARIN, D., DE WITH, P. H. N., AND EFFELSBERG, W. Robust background estimation for complex video sequences. In *Proceedings of International Conference on Image Processing* (2003), vol. 1, pp. 145–148.

[10] GONZALEZ, R. C., AND WOODS, R. E. *Digital Image Processing*, 2nd ed. Addison-Wesley Longman Publishing Co., Inc., 1992.

[11] HARITAOGLU, I., HARWOOD, D., AND DAVIS, L. S. Ghost: A human body part labeling system using silhouettes. In *International Conference on Pattern Recognition* (1998), pp. 77–82.

[12] HARITAOGLU, I., HARWOOD, D., AND DAVIS, L. S. Hydra: Multiple people detection and tracking using silhouettes. *IEEE Workshop on Visual Surveillance* (1999), 280–285.

[13] HUNG, M.-H., PAN, J.-S., AND HSIEH, C.-H. Speed up temporal median filter for background subtraction. In *International Conference on Pervasive Computing, Signal Processing and Applications* (2010), pp. 297–300.

[14] JI, X., AND LIU, H. Advances in view-invariant human motion analysis: A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 40*, 1 (2010), 13–24.

[15] LIU, J., ALI, S., AND SHAH, M. Recognizing human actions using multiple features. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* (2008).

[16] LIU, J., AND SHAH, M. Learning human actions via information maximization. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* (2008).

[17] LV, F., AND NEVATIA, R. Single view human action recognition using key pose matching and viterbi path searching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (2007), pp. 1–8.

[18] MCIVOR, A. M. Background Subtraction Techniques. In *Proc. of Image and Vision Computing* (Auckland, New Zealand, 2000), pp. 147–153.

[19] ORRITE, C., MARTÍNEZ, F., HERRERO, E., RAGHEB, H., AND VELASTIN, S. Independent viewpoint silhouette-based human action modelling and recognition. In *The 1st International Workshop on Machine Learning for Vision-based Motion Analysis* (2008), MLVMA'08.

[20] RABINER, L. R. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of IEEE* (1989), vol. 77, pp. 257–286.

[21] SULAIMAN, S., HUSSAIN, A., TAHIR, N., SAMAD, S., AND MUSTAFA, M. Human silhouette extraction using background modeling and subtraction techniques. In *Information Technology Journal* (2008), vol. 7, pp. 155–159.

[22] WEINLAND, D., BOYER, E., AND RONFARD, R. Action recognition from arbitrary views using 3d exemplars. In *Proc. of the International Conference on Computer Vision* (2007), pp. 1–7.

[23] WEINLAND, D., ÖZUYSAL, M., AND FUA, P. Making action recognition robust to occlusions and viewpoint changes. In *Proceedings of the European Conference on Computer Vision Conference on Computer Vision, Part III* (2010), pp. 635–648.

[24] WEINLAND, D., RONFARD, R., AND BOYER, E. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding 104*, 2, 249–257.

[25] YAN, P., KHAN, S. M., AND SHAH, M. Learning 4D action feature models for arbitrary view action recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* (2008), pp. 1–7.