UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

RECONOCIMIENTO DE ACTIVIDADES EN VIDEO BASADO EN EJEMPLOS

-PROYECTO FIN DE CARRERA-

Pablo Manuel Herranz Fernández

MAYO 2010

RECONOCIMIENTO DE ACTIVIDADES EN VIDEO BASADO EN EJEMPLOS

AUTOR: Pablo Manuel Herranz Fernández
TUTOR: Pedro Tomé González
PONENTE: Javier Ortega García

ATVS Grupo de Reconocimiento Biométrico
(http://atvs.ii.uam.es)
Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
MAYO 2010

Resumen

Las aplicaciones de reconocimiento de acción sobre videos reales requieren el desarrollo de los sistemas que sean rápidos, que puedan manejar una gran variedad de acciones sin el conocimiento a priori del tipo de acciones, que necesiten un número mínimo de parámetros, y que la etapa de estudio sea tan corta como fuera posible. En este trabajo seguiremos tal enfoque. Consideramos las actividades dinámicas como objetos de larga duración temporal, que se caracterizan por rasgos espacio-temporales en múltiples escalas temporales. Basado en esto, diseñamos una medida simple de distancia estadística entre las secuencias de vídeo que capturan las similitudes en su contenido conductual. Esta medida es no paramétrica y por lo tanto puede manejar una amplia gama de complejas acciones dinámicas. Teniendo una medida de distancia basada en el comportamiento entre secuencias, lo usamos para varias de tareas, incluyendo: la indexación de vídeo, la segmentación temporal, y clustering de vídeo basado en acciones. Estas tareas son realizadas sin el conocimiento previo de los tipos de acciones, sus modelos, o sus grados temporales.

Palabras clave:

Acción, movimiento, distancia conductual, secuencia de vídeo, gradiente, segmentación, reconocedor automático de actividades.

Abstract

The action recognition applications on real videos need the development of fast systems, which can handle a variety of actions without a previous knowledge of the type of action, requiring a minimum number of parameters, and a learning stage as short as possible. In addition, this work will continue focus on this approach. We understand dynamic activities as long-term temporal objects, which are characterized by spatio-temporal features at multiple temporal scales. Based on this, we design a simple statistical distance measure in video sequences which capture the similarities in their behavioural content. This measure is non-parametric and therefore, it can handle a wide range of complex dynamic actions. Having a distance measure based on behaviour sequences, is useful to several task as for example: video indexing, temporal segmentation, and action-based video clustering. These tasks are performed without previous knowledge of the types of actions, their models, or their temporary degrees.

Key words:

Action, movement, behavior distance, video stream, gradient, segmentation, automatic activitys recognition.

Agradecimientos

Después de mucho trabajo y tiempo invertido en la realización de este proyecto, es una tarea complicada agradecer a toda la gente que con su presencia a lo largo de este periodo me ha ayudado a llegar a este punto. En primer lugar tengo que agradecer a mi ponente Javier Ortega su confianza al darme la oportunidad de poder formar parte del grupo de investigación ATVS y su apoyo recibido.

Por supuesto tengo que agradecer a Ivana Mikic por ser la persona que hizo posible que este proyecto comenzara y por todos los conocimientos que me transmitió que fueron fundamentales para la finalización del mismo. A mi tutor Pedro Tomé, que recogió el testigo de Ivana, por su gran trabajo, paciencia y dedicación.

También agradezco a todo el profesorado de la Escuela Politécnica Superior por su entrega profesional y personal y por saber guiarnos a todos hacia lo que ahora somos. Al resto de compañeros del ATVS por el buen ambiente generado en el grupo.

Tengo que agradecer a mi familia al completo, abuelos, tíos y a mis padres Eugenia y Manuel, por todo su cariño, apoyo, ánimo, compresión y tantas cosas que me han dado durante toda mi vida que no se pueden describir con palabras.

Pero todo en esta etapa universitaria no han sido estudios, sino más bien desarrollo personal, por eso tengo que agradecer enormemente a todos mis compañeros que ha compartido este largo camino conmigo durante todos estos años, me han dado muchos momentos buenos, risas, felicidad, pero sobre todo lo que más valoro es que me han dado su amistad. Gracias Gus, Kiko, Chus, Peter (los superpepos), Sergio, Vero, Ele, Moni, Esther, Sonso, Imanol, Nit, Mario, Marcos (no sé si me dejo a alguien)… espero que sigamos siempre juntos.

Quiero recordad de forma especial a Guido un gran compañero y amigo que está presente en nuestros corazones. Siempre te echaremos de menos.

Por último quiero agradecer de todo corazón a mis amigos de siempre, con los que he compartido ya tantos años y experiencias que han llegado a convertirse en parte fundamental en mi vida, los momentos con vosotros siempre han sido y serán importantes parara mí. Gracias por estar ahí Valle Nacho, Nico, Ángela, Gonzalo, Javi, y también Elena, María, Carlos… Gracias por todo.

A todos muchas gracias.

Índice de contenidos:

Índice de ilustraciones:

Índice de Tablas:

1. Introducción

1.1. Motivación

Un ser humano cuando observa una determinada imagen o un vídeo tiene la capacidad de reconocer de manera sencilla e inmediata las distintas actividades que se desarrollan en la escena, pero este reconocimiento que resulta fácil de realizar para un humano es un gran desafío para los sistemas encargados del procesamiento de imágenes y vídeo. El reconocimiento automático de las actividades dinámicas es necesario en varios sistemas de análisis de video, como por ejemplo sistemas de seguridad, espacios inteligentes, video indexado basado en acciones, browsing, clustering y segmentación.

Hasta ahora muchos de los trabajos anteriores se centran en reconocer una serie de acciones predefinidas. Estos métodos proponen aproximaciones para capturar las características importantes de las acciones mediante modelos paramétricos especializados, que por lo general, dan lugar al reconocimiento de alta calidad de las acciones estudiadas. Su construcción, generalmente requiere una fase de aprendizaje muy amplia, donde se proporcionan muchos ejemplos de la acción estudiada. Sin embrago, cuando tratamos con aplicaciones del mundo real a menudo este tipo de sistemas presentan grandes problemas, ya que es improbable que se restrinjan para el reconocimiento de acciones modelo cuidadosamente estudiadas. Esto se debe a que cuando tratamos con datos generales de vídeo no hay ningún conocimiento previo sobre los tipos de acciones en la secuencia de vídeo, su grado temporal y espacial, o su naturaleza (periódica / no periódica).

Por este motivo, este trabajo no sólo se restringe al reconocimiento de las acciones cuidadosamente estudiadas, sino que tratamos con datos generales de vídeo en los que a menudo no hay ningún conocimiento previo sobre los tipos de acciones en la secuencia de vídeo, su grado temporal y espacial, o su naturaleza (periódica/no periódica). En este proyecto presentaremos un reconocedor de actividades, para datos de vídeo generales, utilizando una simple medida de distancia conductual entre la acciones que aparecen en una secuencia de vídeo y las actividades ejemplo introducidas en una etapa breve de estudio previo.

1.2. Objetivos

El principal objetivo en este proyecto es conseguir un sistema reconocedor de acciones humanas en vídeo sin tener conocimiento a priori del tipo de actividades que aparecen en la secuencia, basándose únicamente en los datos de un conjunto de ejemplos de movimientos guardados en una fase de estudio previa en nuestra base de datos. El diseño del algoritmo permite a la computadora, en una primera etapa, aprender de forma sencilla y breve modelos de acciones humanas, y más adelante dado un nuevo vídeo, el algoritmo debe ser capaz de decidir si las acciones humanas aprendidas anteriormente están o no presentes en la secuencia.

Esto se consigue mediante una medida de distancia estadística entre las actividades que surgen en las secuencias de vídeo que está basada sólo en el comportamiento. Es decir conserva las variaciones temporales, mientras es insensible a cambios de aspecto como la variación de la ropa, condiciones de iluminación, etc. Esta medida es no paramétrica pudiendo manejar así una amplia gama de comportamientos dinámicos, permitiéndonos así el análisis general de información de vídeo que contiene tipos de acción desconocidos.

Un uso deseado, podría ser detección de actividades inusuales en un video de seguridad en un lugar público, como podría ser un aeropuerto. Otro uso podría ser para el usuario que ve una película, indicar un punto interesante de vídeo que contiene una acción de interés, requiriendo posteriormente al sistema avanzar rápido al siguiente fragmento en el que ocurran acciones similares. Otra aplicación deseada es la segmentación temporal basada en comportamientos de secuencias largas de vídeo. Dada una secuencia larga de video que contiene variedad de acciones nos interesaría detectar los puntos de comienzo y final de las acciones, sin requerir cualquier conocimiento previo de los tipos de acciones o sus grados temporales.

Los objetivos principales serán:

1. **Reconocimiento de actividades basado en ejemplos**

Dado un ejemplo de un fragmento simple de una acción de interés y un amplio conjunto de secuencias, deseamos detectar acciones similares a la acción ejemplo, no teniendo información previa en el contenido del conjunto de vídeos. Esto es logrado mediante comparación de acciones de interés contra todas las subsecuencias de un video, con la misma longitud temporal que el fragmento de ejemplo (similar a la acción de detección). Las subsecuencias con una pequeña distancia a la dada en el fragmento ejemplo son detectadas como una representación de la misma acción. Cuando una

acción se repite múltiples veces en subsecuencias consecutivas será detectada como la misma acción, así el resultado final de la detección, puede incluir segmentos de video de varias longitudes.

2. <u>Reconocimiento de actividades inusuales</u>

Dado ejemplos de actividades conocidas, deseamos detectar acciones suficientemente diferentes. La medida que planeamos diseñar debe ser capaz de distinguir entre las actividades que son suficientemente diferentes y, por tanto, nos permite diseñar un algoritmo para detectar actividades diferentes del conjunto de las actividades presentadas conocidas por el sistema.

1.3. Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Capítulo 1. Introducción:** Introducción, objetivos y motivación del proyecto.

- **Capítulo 2. Fundamentos de la biometría:** Se introducirán los conceptos básicos de la biometría, haciendo referencia a los rasgos biométricos, a la estructura general de los sistemas, a los modos de operación de los sistemas biométricos, etc.

- **Capítulo 3. Estado del arte del reconocimiento de acciones:** Se introducirán conceptos pertenecientes a la teoría de imagen y de vídeo, una visión de los distintos enfoques en reconocimiento movimiento. Posteriormente se hace un estudio de distintos sistemas de reconocimiento existentes.

- **Capítulo 4. Diseño del proyecto:** Se describirán los medios disponibles para el desarrollo, el sistema general y las distintas fases que lo componen.

- **Capítulo 5. Desarrollo del proyecto:** Se describirá el procedimiento seguido para la implementación del sistema y el funcionamiento de cada uno de los componentes desarrollados.

- **Capítulo 6. Experimentos y resultado:** Se presentaran los datos obtenidos tras la evaluación del sistema y se analizaran los resultados conseguidos.

- **Capítulo 7. Conclusiones y trabajo futuro.**

2. Fundamentos de la biometría

2.1. Introducción a la biometría

Existen tres grandes modalidades de reconocimiento: basado en algo que el individuo **sabe** (p.e. una contraseña), en algo que **tiene** (p.e una llave) o en algo que **es** (p.e su cara). El reconocimiento biométrico se corresponde a este último grupo.

El término biometría viene del griego "bio" que significa vida y "metría" que significa medida o medición. La biometría es el conjunto de métodos automatizados que analizan determinadas características humanas para identificar o autentificar personas.

En los distintitos sistemas de reconocimiento biométrico se utilizan diferentes rasgos para conseguir llegar a la identificación y autentificación del individuo. Todos estos rasgos tienen que cumplir en mayor o menor medida estas características:

- Universalidad.
- Unicidad.
- Permanencia o estabilidad.
- Mensurabilidad o evaluabilidad.
- Aceptabilidad.
- Rendimiento.
- Evitabilidad o fraude.

2.1.1. Rasgos biométricos

Desafortunadamente no hay ningún rasgo biométrico que cumpla todas las características anteriores con total satisfacción. Por lo tanto la elección del rasgo que se utilizara para cada aplicación depende de las características del mismo y de los requisitos de la aplicación. Los rasgos biométricos con los que se trabaja son los siguientes.

ADN: El ADN tiene como factor positivo que es único para cada individuo. En cambio, como factores negativos se puede destacar que es un rasgo biométrico fácil de robar, el proceso de reconocimiento es lento y es necesario que sea asistido por un persona y además el ADN puede revelar características del individuo que no quiere hacer públicas.

***Oreja*:** Para este tipo de reconocimiento se emplea la forma del borde y de las estructuras gelatinosas que la componen.

***Cara*:** Es el rasgo biométrico que más comúnmente se utiliza para el reconocimiento humano entre individuos junto con la voz. Como aspectos positivos se puede decir que es un rasgo muy aceptado y se adquiere mediante un método no invasivo. Por otro lado se puede engañar al sistema mediante el uso de mascaras, luego el sistema debería estar preparado para posibles cambios debidos a la edad, iluminación, expresión, etc.

***Termogramas*:** El calor que radia cada individuo es característico. Este rasgo se captura de forma no intrusiva mediante cámaras de infrarrojos. Las desventajas son el coste de los sensores y la vulnerabilidad frente a fuentes de calor no controlables.

***Venas de la mano*:** La identificación mediante las venas de la mano es muy segura y difícilmente falsificable debido a las múltiples características que los diferencian, por ello está modalidad biométrica es considerada más segura que otras.

***Huellas dactilares*:** La identificación mediante huella dactilar se lleva utilizando muchos siglos. Una huella es única para cada individuo y dentro de cada individuo para cada dedo. Actualmente la exactitud de estos sistemas es muy elevada y están muy extendidos.

***Forma de caminar*:** Este es un rasgo biométrico complejo a nivel espacio-temporal. No es muy distintivo, pero puede ser suficientemente discriminatorio si por ejemplo los usuarios están obligados a pasar por un mismo sitio. Pertenece a los rasgos biométricos de comportamiento y varía a lo largo del tiempo, pero se adquiere de forma no invasiva con una cámara.

***Geometría de la mano*:** Este rasgo no es muy distintivo y está sujeto a cambios a lo largo de la vida. Se basa en un conjunto de medidas significativas como la forma de la mano, el tamaño de la palma, la longitud y el ancho de los dedos.

***Iris*:** Es un rasgo altamente discriminatorio y único para cada uno de los ojos del individuo. En la antigüedad de una gran participación por parte del usuario en el proceso de captura, pero hoy en día es posible su captura de manera poco invasiva. Debido a la complejidad de los sistemas la tecnología sigue siendo clara.

***Huella palmar*:** Es un rasgo similar a la huella dactilar, pero con mayor información, ya que posee mayor área, luego es por esto que se considera más distintivo.

***Olor*:** Una parte del olor que emiten los humanos es distintivo, el olor puede ser capturado por sensores químicos, pero es complicado separarlo de sustancias artificiales.

***Escáner de retina*:** La estructura vascular de la retina es distinta para cada ojo y para cada individuo. Debido a la complejidad de su captura es un rasgo difícilmente falsificable.

***Firma*:** La firma es característica de cada persona. Este método se usa habitualmente en multitud de transacciones como sistema de autentificación. La firma tiene una amplia variabilidad a lo largo del tiempo y también con aspectos físicos y emocionales del individuo.

***Voz*:** La voz es una combinación de características físicas y de conducta, las físicas no varían a lo largo del tiempo, pero las de conducta se ven alteradas por múltiples factores. Este rasgo puede ser fácilmente imitado, pero a su vez está muy aceptado y es fácil de conseguir.

***Escritura*:** La escritura no es muy distintiva y varía a lo largo del tiempo, pero su captura es poco invasiva.

***Dinámica de tecleo*:** Cada persona tiene una dinámica de tecleo característica. Este rasgo es de conducta por lo que varía a lo largo del tiempo y es poco distintivo, pero proporciona información suficientemente discriminatoria para identificación en casos sencillos. Para su captura basta con emplear secuencias de tecleo del usuario, por lo que no es intrusivo.

2.1.2. Aplicaciones de los sistemas biométricos

Un sistema biométrico consiste en un reconocedor de parámetros que funciona siguiendo los siguientes pasos: captura el rasgo biométrico, extrae sus características y las compara con los patrones almacenados en la base de datos para decidir la identidad del individuo.

Las aplicaciones de los sistemas biométricos se pueden agrupar de la siguiente forma:

- Aplicaciones comerciales: protección de datos electrónicos, protección de red, e-comercio, cajeros automáticos, control de acceso físico, etc.
- Aplicaciones gubernamentales: DNI, carnet de conducir, pasaporte, control de fronteras, etc.
- Aplicaciones forenses: identificación de cadáveres, investigación criminal, determinación de parentesco, etc.

2.1.3. Problemas y limitaciones de los sistemas biométricos

Sin embargo los sistemas biométricos también tienen problemas y limitaciones que pueden variar los rasgos biométricos, las causas más comunes que provocan esto son:

***Presentación inconsciente*:** La señal capturada por el sensor depende tanto de las características del rasgo biométrico como de la forma que se presenta dicho rasgo.

***Presentación irreproducible*:** El hecho de que se produzca una variación de la señal capturada en diferentes adquisiciones puede ser debido a que como los rasgos biométricos representan una característica biológicas o de comportamiento los accidentes o los adornos pueden cambiar su estructura o su aspecto exterior.

***Captura imperfecta*:** Las condiciones de captura de una señal en la práctica no son perfectas y causan variaciones en la señal capturada.

***Ruido en los datos adquiridos*:** El ruido puede estar producido por las condiciones del sensor o condiciones ambientales desfavorables. Las adquisiciones con ruido pueden provocar errores en la identificación.

***Unicidad*:** Puede existir similitudes entre diferentes usuarios en las características usadas para representar un rasgo biométrico. Esto disminuye la capacidad de discriminar mediante este rasgo.

***No universalidad*:** Puede ser que un subconjunto de individuos carezca de un rasgo biométrico que se espera que tengan todos los individuos.

***Ataques*:** Un impostor puede intentar imitar un rasgo biométrico para sortear el sistema.

2.1.4. Aceptación en la sociedad

Es la sociedad la que determina el éxito de los sistemas de identificación basados en rasgos biométricos. La facilidad y comodidad en la interacción con el sistema contribuye a su aceptación. Las tecnologías que requieren muy poca cooperación o participación de los usuarios suelen ser percibidas como más convenientes. Por otro lado, los rasgos biométricos que no requieren la participación del usuario en su adquisición pueden ser capturados sin que el individuo se dé cuenta y esto es percibido como una amenaza a la privacidad por parte de muchos usuarios. El tema de la privacidad adquiere gran relevancia con los sistemas de reconocimiento biométrico porque los rasgos biométricos pueden proporcionar información muy personal de un individuo, como afecciones médicas, y esta información puede ser utilizada de forma poco ética.

Por otro lado, los sistemas biométricos pueden ser empleados como uno de los medios más efectivos para la protección de la privacidad individual. Si un individuo extravía su tarjeta de crédito y otra persona la encuentra, podría hacer un uso fraudulento de ella. Pero si la tarjeta de crédito únicamente pudiese ser utilizada si el impostor suplantase los rasgos biométricos del usuario, éste estaría muchísimo más protegido. Otra ventaja del uso de los rasgos biométricos consiste en limitar el acceso a información personal.

La mayoría de los sistemas biométricos comerciales disponibles hoy en día no almacenan las características físicas capturadas en su forma original, sino que almacenan una representación digital en un formato encriptado. Esto tiene dos propósitos: el primero consiste en que la característica física real no pueda ser recuperada a partir de su representación digital, lo que asegura privacidad, y el segundo se basa en que el encriptado asegura que sólo la aplicación designada puede usar dicha representación digital.

2.2. Sistemas automáticos de Reconocimiento

2.2.1. Estructura

Los sistemas automáticos de reconocimiento de patrones poseen una estructura funcional común, formada por varias fases, cuya forma de proceder depende de la naturaleza del rasgo o señal a reconocer. La primera etapa corresponde a la zona a la cual el usuario tiene acceso, es decir, el interfaz de usuario, en ella cabe destacar el sensor que es el encargado de capturar el rasgo biométrico con el que pretende hacer el reconocimiento. La siguiente etapa corresponde al sistema los módulos más destacables son el módulo de extracción de características, el comparador y el decisor estos módulos son básicos en cualquier sistema de reconocimiento, el resto de módulos sirven para procesar la señal y pueden ser opcionales.

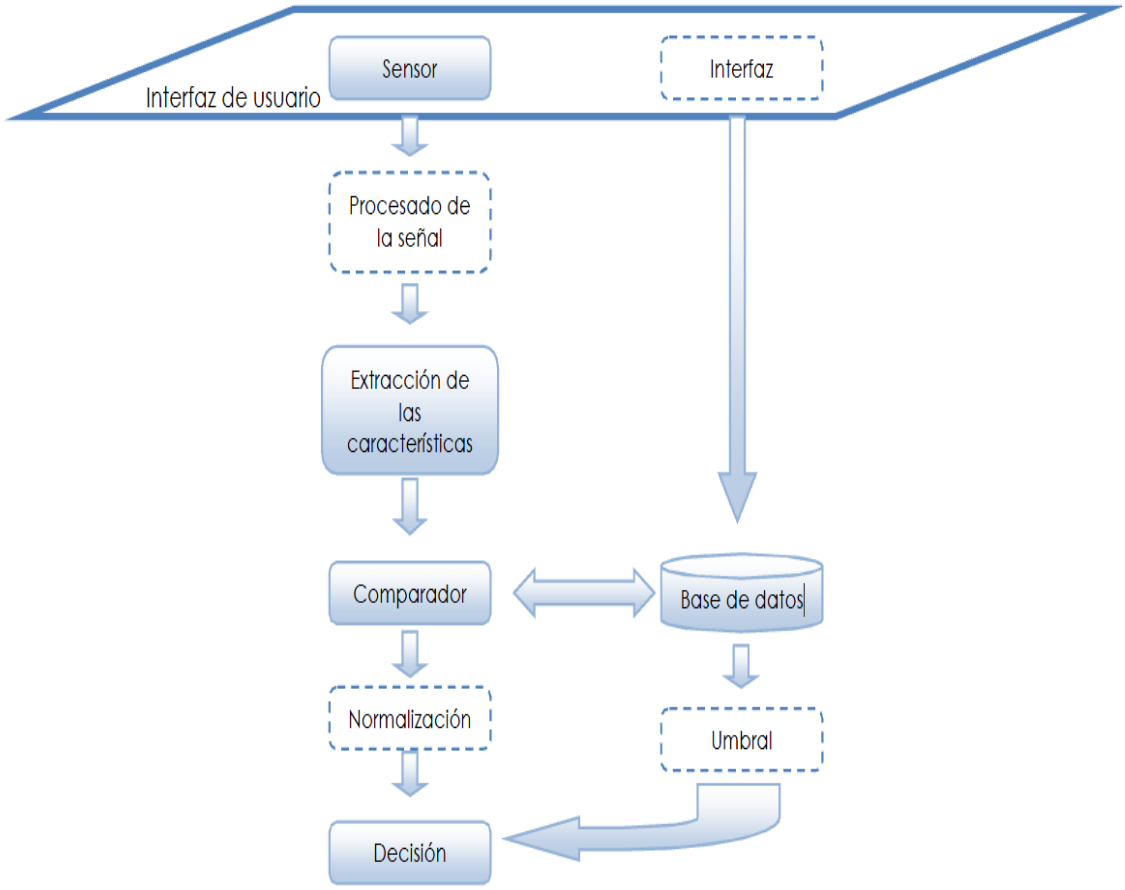


Figura 2.1: Esquema de funcionamiento de un sistema de reconocimiento biométrico.

Para explicar más en profundidad el diagrama anterior lo podemos dividir en cuatro fases de funcionamiento que son la adquisición de datos, seguida del preprocesado, extracción de características y terminando con la comparación de patrones.

Adquisición de datos: En esta fase un sensor se encarga de recoger los datos analógicos y los convierte a formato digital. Esta fase es de suma importancia ya que de ella depende la cantidad y la calidad de información que se introduce en el sistema, es decir, la información con la que van a trabajar las siguientes fases, y por lo tanto de esta fase depende mucho el resultado final que se obtiene.

Preprocesado: En algunos casos es necesario acondicionar la información capturada para eliminar posibles ruidos o distorsiones producidas en la fase de adquisición, o para normalizar la información a unos rasgos específicos para tener una mayor efectividad en el reconocimiento posterior.

Extracción de características: En esta fase se extraen aquellas características que sean discriminantes entre distintos individuos y que al mismo tiempo permanezcan invariables para un mismo usuario, eliminando la información que no resulte útil en el proceso de reconocimiento. Con esto se reduce la duración de todo el proceso, su coste computacional y el espacio para almacenar la plantilla.

Comparación de patrones: En esta fase se comparan las características anteriormente extraídas con los diferentes modelos almacenados en la base de datos con el fin de evaluar de la correspondencia entre los patrones de entrada y el modelo particular que queremos reconocer.

Dentro del ámbito de sistemas de reconocimiento biométricos, se puede trabajar de dos formas: modo reconocimiento positivo o modo reconocimiento negativo. El reconocimiento positivo trata de determinar si un usuario es quien afirma ser, mientras que el reconocimiento negativo intentan determinar si un usuario es quien niega ser, un ejemplo de este tipo de sistemas está instalado en los aeropuertos, con las listas negras de terroristas. Cabe destacar que la identificación negativa sólo puede ser realizada mediante rasgos biométricos, y no mediante métodos clásicos como contraseñas o llaves.

2.2.2. Modos de operación

Un sistema de reconocimiento consta de tres modos de operación primeramente una fase previa que sería el modo registro, seguidamente de dos posibles modos de funcionamiento que son verificación o identificación.

En el modo registro, el sistema adquiere una plantilla del rasgo biométrico. Durante esta etapa, se produce un preprocesado de la señal biométrica, se extraen las características de interés y se almacenan en el sistema construyendo un modelo. Dependiendo de la aplicación esta información será guardada en la base de datos del sistema o en otro tipo de dispositivos externos.

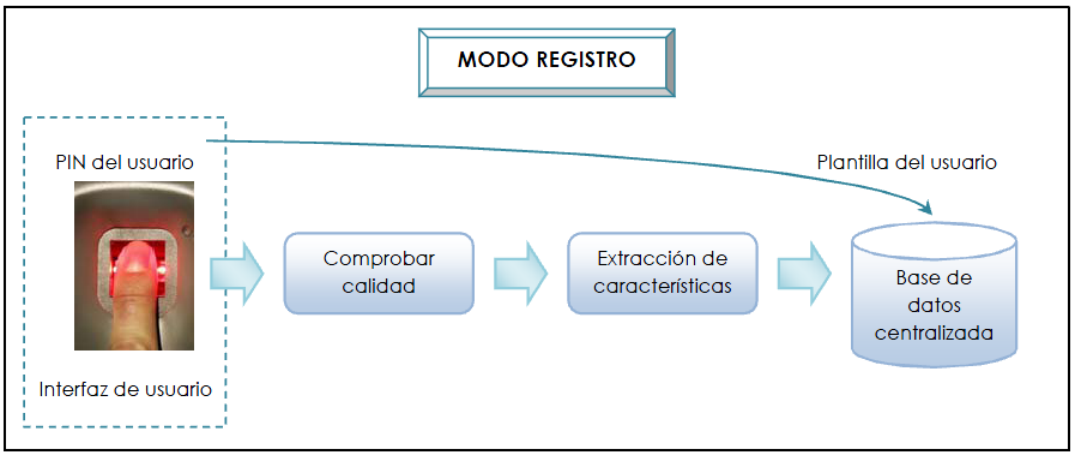


Figura 2.2: Esquema de funcionamiento en modo registro.

En el modo verificación o autenticación el sistema valida la identidad de una persona comparando el rasgo biométrico capturado en la entrada con su propia plantilla biométrica previamente almacenada en la base de datos.

Las dos posibles salidas en este modo de funcionamiento (verdadero/ falso) dan lugar a la aparición de dos errores distintos:

- ***Falso Rechazo:*** se produce cuando el sistema indica que la información adquirida del usuario en la entrada no se corresponde con la plantilla almacenada, cuando realmente sí se corresponde.

- ***Falsa Aceptación:*** se produce cuando el sistema indica que la información adquirida del usuario en la entrada sí se corresponde con la plantilla almacenada, cuando realmente no se corresponde.

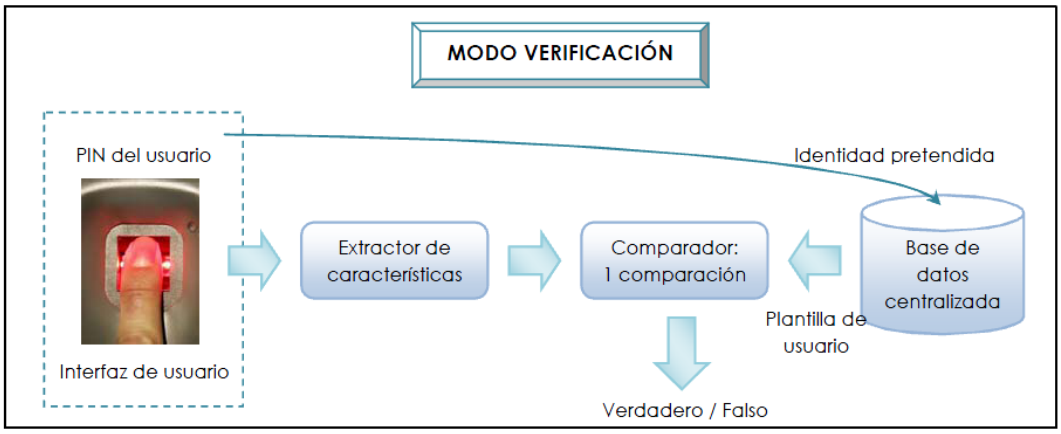


Figura 2.3: Esquema de funcionamiento en modo verificación.

En el modo identificación el sistema recibe un rasgo biométrico determinado y comprueba se corresponde con alguno de los modelos almacenados en la base de datos. El sistema tratara de decidir si el usuario esta o no en la base de datos. El coste computacional de este modo es muy elevado.

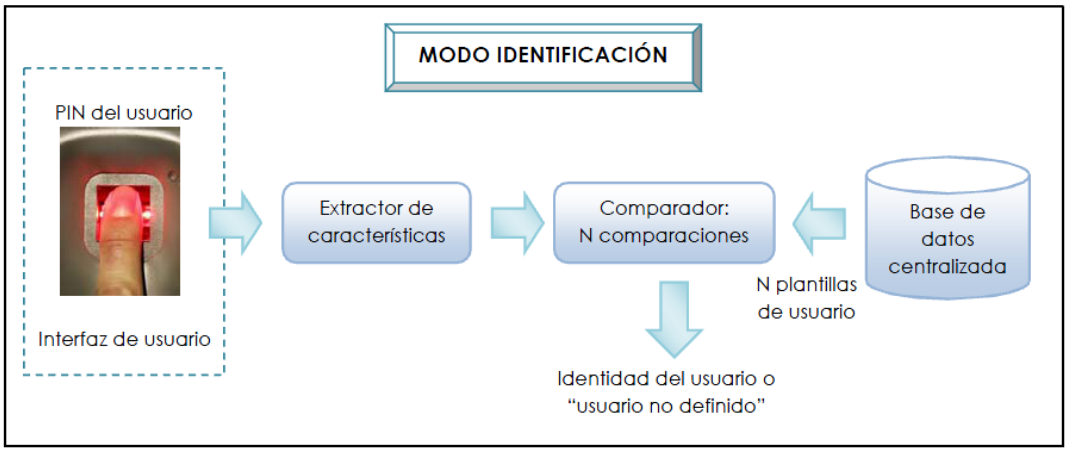


Figura 2.4: Esquema de funcionamiento en modo identificación.

2.2.3. Evaluación del rendimiento

Para determinar la capacidad y la bondad de nuestro sistema de reconocimiento necesitamos una medida objetiva del rendimiento del mismo. Entendemos como rendimiento de un sistema biométrico como la precisión en el proceso de reconocimiento.

En cualquier sistema biométrico hay que tener en cuenta que dos muestras de un mismo rasgo no son exactamente iguales debido a imperfecciones en las condiciones en las que se captura la imagen, cambios fisiológicos o de comportamiento, factores ambientales y

a la interacción del usuario con el sensor entre otros. Por tanto la respuesta del comparador de un sistema biométrico consiste en una puntuación que cuantifica la similitud entre la entrada y el patrón de la base de datos con el que se está comparando. Cuanto mayor sea la similitud entre muestras mayor será la puntuación devuelta por el comparador.

La decisión del sistema está regulada por un umbral: los pares de muestras que generen puntuaciones mayores o iguales que el umbral se supondrán correspondientes a la misma persona, mientras que los pares de muestras cuya puntuación sea inferior al umbral establecido se considerarán de personas diferentes.

2.2.4. Sistemas multibiométricos

Los sistemas multibiométricos son aquellos que combinan varias fuentes de información biométrica para mejorar el rendimiento de un determinado sistema, se mejora la seguridad del sistema al aumentar la dificultad de imitar o falsificar varios rasgos simultáneamente.

Un sistema de este tipo puede operar de tres modos diferentes:

- ***Modo serie:*** las salidas del análisis de un rasgo biométrico se usan como entrada para análisis del siguiente rasgo, reduciendo así en cada paso el número de identidades posibles antes de emplear la siguiente característica.

- ***Modo paralelo:*** la información de múltiples rasgos biométricos se emplea simultáneamente en el proceso de reconocimiento.

- ***Modo jerárquico:*** los clasificadores individuales se combinan en una estructura de árbol.

Los sistemas biométricos multimodales pueden trabajar en cinco posibles escenarios:

- Múltiples sensores
- Múltiples rasgos
- Múltiples instancias de un mismo rasgo
- Múltiples capturas de un mismo rasgo
- Múltiples representaciones/comparaciones para un mismo rasgo

3. Estado del arte

3.1. Teoría de imágenes

El trabajo con imágenes es una parte fundamental dentro del proyecto a realizar, ya que mediante el procesamiento de las mismas se logra la detección de los eventos predeterminados. Es por ello que en este apartado se explicarán las características más importantes que poseen las imágenes.

3.1.1. Definición de pixel

El píxel es la unidad más pequeña en la que se puede descomponer una imagen digital, ya sea una fotografía, un fotograma de vídeo o un gráfico. Los píxeles aparecen como pequeños cuadrados en color, en blanco o en negro, o en matices de gris. Las imágenes se forman como una matriz rectangular de píxeles, donde cada píxel forma un punto diminuto en la imagen total.

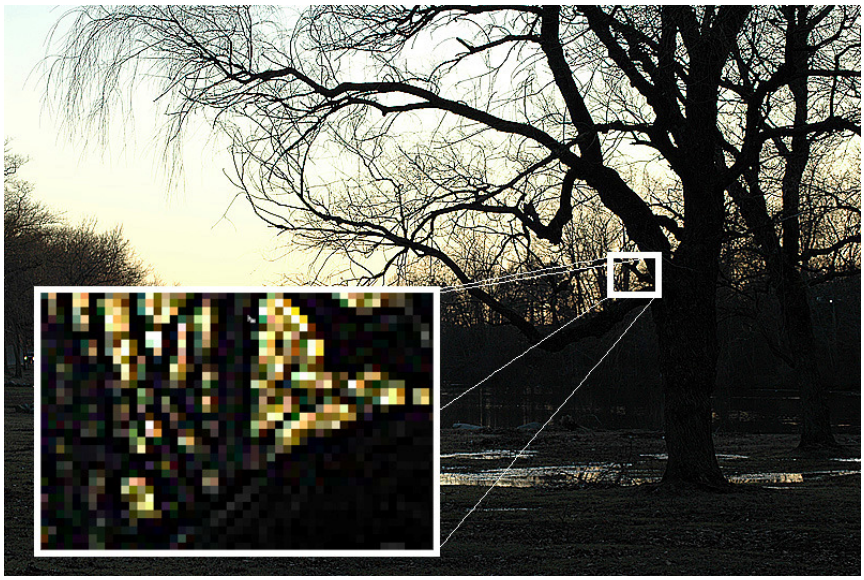


Figura 3.1: Ampliación de un sector de la imagen.

Como se observa en la figura 3.1, cuando se amplifica una pequeña región de la imagen (zoom) se puede apreciar fácilmente los píxeles que conforman dicha imagen como si fueran pequeños cuadraditos. Los píxeles son tan diminutos que a simple vista uno no se puede percatar de su forma cuadrada.

En las imágenes de mapa de bits cada píxel se codifica mediante un conjunto de bits de longitud determinada (la llamada profundidad de color), por ejemplo, puede codificarse un píxel con un byte, u 8 bits, de manera que cada píxel admite 256 variantes (2 dígitos por bit, elevados a la octava potencia, es decir, 2^8). En las imágenes de color verdadero, se suelen usar tres bytes para definir un color, es decir, en total se puede representar 16.777,216 colores diferentes (2^{24} variantes).

Para poder transformar la información numérica que almacena un píxel en un color se debe conocer, además de la profundidad de color (el tamaño en bits del píxel), el modelo de color que se está usando, como por ejemplo el modelo de color RGB. A continuación podemos ver una figura que nos muestra como están distribuidos los pixeles en de una imagen que utiliza el modelo RGB.

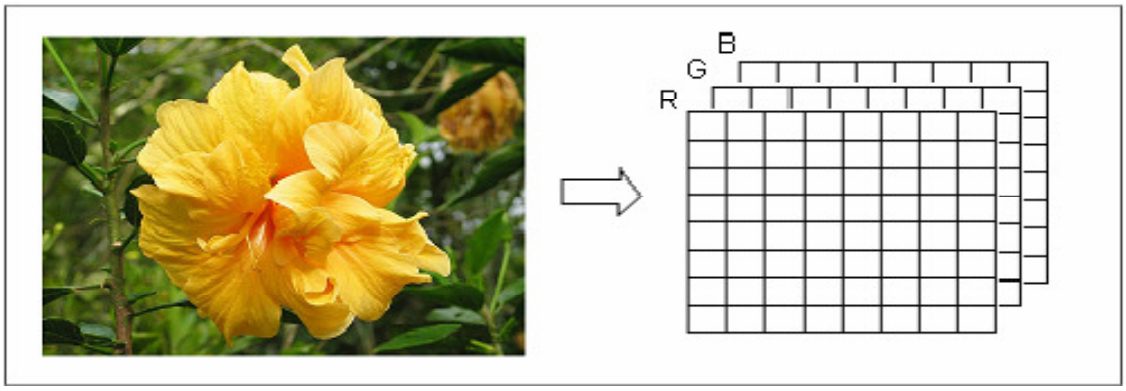


Figura 3.2: Composición de una imagen en RGB.

3.1.2. Modelos de color de un pixel

Algunos de los modelos de color más conocidos que se utilizan en la composición del color de un píxel son, el modelo de color RGB, CMYK y YIQ, los cuales serán descritos a continuación:

El modelo de color RGB (Red-Green-Blue) permite crear un color componiendo tres colores básicos: el rojo, el verde y el azul. En el modelo RGB es frecuente que se use un byte (8 bits) para representar la proporción de cada una de las tres componentes primarias. Así, de una manera estándar, la intensidad de cada una de las componentes se mide según una escala entre 0 y 255. De esta forma, cuando una de las componentes vale 0, significa que esta no interviene en la mezcla y cuando vale 255 (2^8-1) significa que interviene aportando el máximo de ese tono.

Por lo tanto, el rojo se obtiene con (255, 0, 0), el verde con (0, 255, 0) y el azul con (0,0, 255), obteniendo un color resultante monocromático. La combinación de dos colores en

nivel 255 con un tercero en nivel 0 da lugar a tres colores intermedios: el amarillo (255, 255 ,0), el cian (0, 255, 255) y el magenta (255, 0, 255).

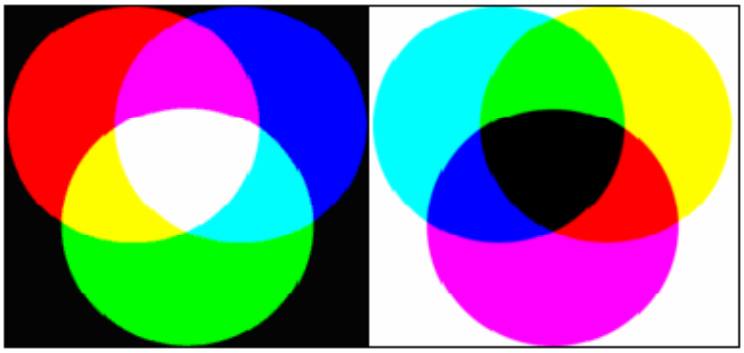


Figura 3.3 Mezcla aditiva de colores del RGB y el modelo sustractivo del CMYK.

El gráfico de la figura 3.3 muestra en su parte izquierda la mezcla aditiva de colores del modelo RGB, en donde la combinación de sus tres componentes primarias compone el color blanco y la ausencia de sus componentes forma el color negro. En la derecha de la figura, se muestra la mezcla sustractiva del modelo CMYK.

Las ecuaciones siguientes muestran la forma de conversión del modelo RGB al modelo CMYK. Para convertir valores entre RGB y CMYK, se debe utilizar un valor CMY intermedio. Estos valores de color se deben representar como un vector, estos pueden variar entre 0.0 (color inexistente) y 1.0 (color totalmente saturado):

$t_{CMYK} = \{C, M, Y, K\}$ es el cuádruplo del valor CMYK entre $[0,1]^4$ (3.1)

$t_{CMY} = \{C, M, Y\}$ es el triple del valor CMY entre $[0,1]^3$ (3.2)

$t_{RGB} = \{R, G, B\}$ es el triple del valor RGB entre $[0,1]^3$ (3.3)

El modelo CMYK (acrónimo de Cian, Magenta, Yellow y Key) es un modelo de colores sustractivo que se utiliza en la impresión a color. Este modelo se basa en la mezcla de pigmentos de los colores Cian (C), Magenta (M), Amarillo (Y) y Negro (K) para crear otros.

La mezcla de colores CMY ideales es sustractiva (al imprimir cian, magenta y amarillo en fondo blanco resulta el color negro). El modelo CMYK trabaja en base a la absorción de la luz. Los colores que se ven son de la parte de la luz que no es absorbida. El cian es el opuesto al rojo, lo que significa que actúa como un filtro que absorbe dicho color (-R +G +B). Magenta es el opuesto al verde (+R -G +B) y amarillo el opuesto al azul (+R +G -B).

El modelo YIQ define un espacio de color, usado antiguamente por el estándar de televisión NTSC. La componente "Y" representa la información de luminancia y es el único componente utilizado por los televisores de blanco y negro. "I" y "Q" representan la información de crominancia o del color. La luminancia (Y) viene dada por el brillo de un objeto, pudiendo producir dos objetos con tonalidades diferentes con la misma sensación lumínica. La señal de luminancia es la cuantificación de esa sensación de brillo.

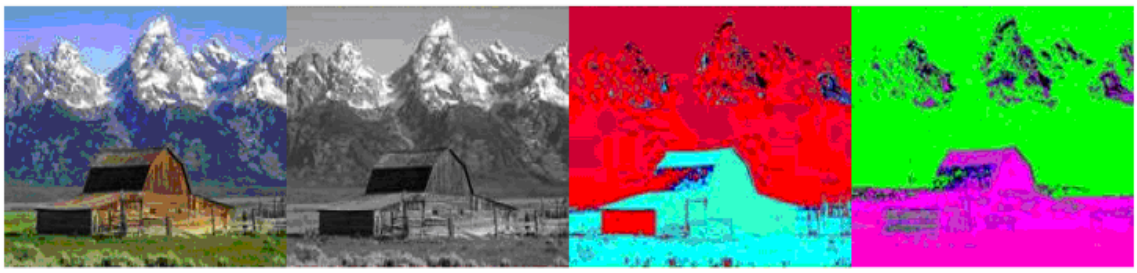


Figura 3.4: De izquierda a derecha, imagen con sus componentes Y, I y Q.

En la figura 3.4, de izquierda a derecha, se muestra primero la imagen en color real, luego se muestra la imagen únicamente con sus valores de luminancia Y (brillo de la imagen). Luego se muestra la imagen con la información de crominancia I (naranja-azul) y la que sigue utiliza los valores de crominancia Q (púrpura-verde). La matriz de conversión del modelo RGB al modelo YIQ se muestra en la siguiente ecuación:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.595716 & -274453 & -321263 \\ 0.211456 & -.522591 & 0.311135 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.4)$$

La representación YIQ se emplea a veces en transformaciones de procesamiento digital de imágenes en color. Por ejemplo, aplicando una ecualización del histograma directamente a los canales en una imagen RGB se alterarían los colores unos en relación con otro, resultando una imagen con colores que no tienen sentido. En vez de ello, si la ecualización el histograma es aplicada al canal Y de la representación YIQ de la imagen, sólo se normalizan los niveles de brillo de la imagen.

3.1.3. Resolución de imágenes

La resolución de imágenes describe cuánto detalle puede observarse en una imagen. El término es comúnmente utilizado en relación a imágenes de fotografía digital, pero también se utiliza para describir la nitidez de la misma. Tener mayor resolución se

traduce en obtener una imagen con más detalle o calidad visual y por tanto mayor información.

Para las imágenes digitales almacenadas como mapa de bits, la convención es describir la resolución de la imagen con dos números enteros, donde el primero es la cantidad de columnas de píxeles y el segundo es la cantidad de filas de píxeles. La convención que le sigue en popularidad es describir el número total de píxeles en la imagen (usualmente expresado como la cantidad de megapíxeles), que puede ser calculado multiplicando la cantidad de columnas de píxeles por la cantidad de filas de píxeles.

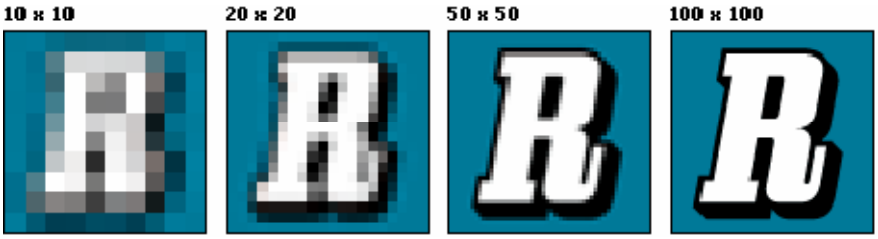


Figura 3.5: De izquierda a derecha, aumento de la resolución de una imagen.

Como se muestra en la figura 3.5, el primer cuadro de la izquierda posee una resolución de 10x10 (10 columnas y 10 filas de píxeles) en la que la imagen es bastante borrosa y como se observa en las imágenes siguientes hacia la derecha, a medida que la resolución aumenta también aumenta la calidad y la nitidez de la imagen.

3.2. Teoría del video

El vídeo es la tecnología de captación, grabación y reproducción de imágenes, que nos permite, al ejecutar estas imágenes en secuencia, simular movimiento. Existe la posibilidad de que las imágenes que componen el vídeo estén acompañadas de sonido. Etimológicamente la palabra video proviene del verbo latino “videre”, y significa "yo veo".

Los comienzos del video están relacionados con el intento de cubrir las necesidades que tenía la televisión. En efecto, las primeras trasmisiones televisivas se realizaban en vivo y con la posibilidad de grabarlas se facilitaba sobremanera el trabajo de programación; en este sentido, los juegos olímpicos de Tokio en el año 1964 fueron el primer caso en que se realiza una transmisión en diferido. Ya a finales de los años setenta se consolida definitivamente como una tecnología independiente de la televisión.

En la actualidad, con la proliferación de medios digitales, el uso de videos ha alcanzado un carácter masivo que difícilmente se haya soñado cuatro décadas atrás. El otro punto importante a considerar es el abaratamiento de los medios tecnológicos de grabación, que se hacen cada día más accesibles.

El vídeo puede presentarse en distintos formatos, como pueden ser cintas de vídeo analógico (VHS, Betamax) o en formatos digitales (DVD, MPEG-4). La calidad del vídeo vendrá determinada por distintos factores, como el método de captura o el tipo de almacenamiento elegido.

3.2.1. Los estándares de vídeo analógico

Existen diversos estándares analógicos. El NTSC (Nacional Televisión System Comité) sigue las especificaciones establecidas en 1950 para la introducción de la televisión en color. Se usa en USA y Japón. El PAL (Phase Alternating Line) es el sistema propio de Australia, Oriente Medio, Asia y Europa con la excepción de Francia que utiliza el SECAM. Ninguno de estos sistemas resulta óptimo para una visualización en la pantalla del ordenador a causa de algunas diferencias técnicas importantes.

Tamaño del fotograma. Mientras que en la pantalla del televisor convencional la imagen se forma en base a líneas horizontales (525 en NTSC y 576 en PAL), en la pantalla del ordenador se crea en base a un mosaico de píxeles. Por otra parte, mientras la resolución del televisor es fija, la de la pantalla de ordenador es variable y en ella los reescalados de la imagen pueden ser necesarios.

Fotogramas por segundo (fps). El vídeo es en esencia una sucesión de fotogramas que al visionarse a una determinada velocidad crea la ilusión de movimiento. En NTSC la frecuencia es de 30 fps mientras que el valor en PAL es de 25. En aplicaciones multimedia es frecuente que el vídeo se reproduzca a la mitad de estos valores.

Píxel Aspect Ratio. Existen diferencias también que atañen a la forma del píxel. Mientras en una pantalla de televisor es rectangular, en un monitor informático es cuadrado. En consecuencia una misma imagen puede aparecer deformada. En función de cuál sea el destino del vídeo será preciso llevar a cabo o no la corrección de este parámetro. Algunas aplicaciones de edición de vídeo llevan a cabo esta compensación al mostrar la imagen en una ventana del monitor. Si el destino final del vídeo es la web es recomendable adaptar los píxeles a unas proporciones de 3x4.

Vídeo entrelazado y progresivo. La imagen de vídeo analógico consiste en dos campos entrelazados cuya suma forma un fotograma. La necesidad del entrelazado proviene de una limitación técnica de los inicios de la televisión y fue la solución que evitaba que se produjera excesivo efecto de parpadeo aún trabajando con una baja frecuencia de imagen. Con el advenimiento del vídeo digital esta característica constituyó un serio obstáculo en la integración de formatos analógicos y digitales y no es descartable que en el futuro desaparezca. De hecho, en los nuevos formatos de televisión de alta definición se ha eliminado el entrelazado.

3.2.2. El paso a vídeo digital

La generalización del vídeo digital se ha producido de la mano del DV y la posibilidad de editar en el ordenador. A la facilidad de uso de las cámaras DV y su notable calidad de imagen se unen las potencialidades que se derivan de la edición digital. A diferencia de los casos en los que la fuente de origen es un vídeo analógico, el trabajar en formato digital en origen elimina la necesidad de disponer de tarjeta digitalizadora en el ordenador. De hecho un simple puerto Firewire, USB o USB-2 permite la transferencia de datos entre la cámara y el equipo informático. El sentido puede ser también el inverso cuando se graba el master ya editado en una cámara o magnetoscopio externo.

La edición digital recupera en cierto modo el espíritu de la edición cinematográfica clásica. Editar un vídeo analógico sobre cinta implica no poder cortar y suprimir secuencias innecesarias, o no poder añadir nuevos planos a una cinta ya editada. Es lo que denominamos edición lineal que se contrapone a la clásicamente ejercida en el cine. Mientras en éste se corta físicamente el film, se suprimen fotogramas o se añaden libremente, en la edición analógica del vídeo el mismo procedimiento es imposible. Uno de los grandes cambios aportados por la edición digital es la recuperación de la no linealidad. Trabajar en formato digital permite insertar, suprimir, aplicar efectos, sumar capas,…, sin perder en absoluto calidad. En teoría son posibles infinitas generaciones a través de la exportación y reimportación de los clips a un proyecto.

La introducción del ordenador como instrumento de edición de vídeo se produjo en los entornos profesionales los últimos años del siglo pasado y se generalizó para el público en general los primeros de éste. Los procesos de edición digital se han generalizado y si bien existen diversas posibilidades de software para llevarlos a cabo puede afirmarse que forman un conjunto de procedimientos y tareas bastante uniforme. En cambio el uso del vídeo editado es ya todo otro tema. Por decirlo de algún modo, un campo que se diversifica y ramifica ampliamente.

3.2.3. Usos del vídeo digital

En términos generales podemos hablar de dos grandes tipos de salidas y usos.

- El primer grupo lo integran las salidas para teledifusión y soportes que como el DVD o las cónsolas de videojuegos no plantean problema por gestionar grandes volúmenes de información. En general prima en ellos una elevada calidad de imagen, pero quizás el común denominador que más nos interesa ahora es que en ninguno existen problemas derivados del peso de los archivos. Que el material de vídeo ocupe gigas no reviste mayor importancia que la de disponer de un equipo con las prestaciones adecuadas. Una vez realizada la edición y guardado el master, ya sea en cinta, ya sea en un soporte óptico, los archivos de trabajo se borran del disco duro.

- El segundo grupo se refiere a la salida para multimedia, la web o los dispositivos móviles. En él se incluyen los clips destinados a ser reproducidos en un ordenador, ya sea a través de un soporte óptico o sea a través de la web, y los destinados a dispositivos como los teléfonos móviles o las PDAs.

3.2.4. Codecs y métodos de compresión

Códec es una abreviatura de Codificador-Decodificador. Describe una especificación desarrollada en software, hardware o una combinación de ambos, capaz de transformar un archivo con un flujo de datos o una señal. Los códecs pueden codificar el flujo o la señal (a menudo para la transmisión, el almacenaje o el cifrado) y recuperarlo o descifrarlo del mismo modo para la reproducción o la manipulación en un formato más apropiado para estas operaciones. Los códecs son usados a menudo en videoconferencias y emisiones de medios de comunicación.

Un códec de video es un tipo de códec que permite comprimir y descomprimir video digital. Normalmente los algoritmos de compresión empleados conllevan una pérdida de información.

El problema que se pretende acometer con los códec es que la información de video es bastante ingente en relación a lo que un ordenador normal es capaz de manejar. Es así como un par de segundos de video en una resolución apenas aceptable puede ocupar un lugar respetable en un medio de almacenamiento típico (disco duro, CD, DVD) y su

manejo (copia, edición, visualización) puede llevar fácilmente a sobrepasar las posibilidades de dicho ordenador o llevarlo a su límite.

La finalidad de los códec es obtener un almacenamiento substancialmente menor de la información de vídeo. Esta se comprime en el momento de guardar la información hacia un archivo y se descomprime, en tiempo real, en el momento de la visualización. Se pretende, por otro lado, que éste sea un proceso transparente para el usuario, es decir, que éste no intervenga o lo haga lo menos posible.
Existe un complicado equilibrio entre la calidad de video, la cantidad de datos necesarios para representarlo (también conocida como tasa de bits), la complejidad de los algoritmos de codificación y decodificación, la robustez frente a las pérdidas de datos y errores, la facilidad de edición, la posibilidad de acceder directamente a los frames, y otros factores.

Básicamente existen dos métodos de compresión, la denominada compresión espacial y la temporal. En la espacial se reduce la información comprimiendo la existente en el interior de cada frame. En lugar de describir la imagen píxel a píxel, señalando por ejemplo la posición y color de los píxeles, el códec de compresión generaliza describiendo área similares y sus características de luz y color. Así por ejemplo, en lugar de reproducir un cielo azul píxel a píxel se describiría el mismo como un área con características de luz y color similares. Una conclusión clara es que cuantos menos detalles variados presente una imagen más fácilmente el códec podrá generalizar y comprimir. Crear vídeos con fondos simples facilita la compresión y la reducción, del mismo modo que trabajar con trípode en lugar de cámara en mano supone estabilizar los fondos y por lo tanto facilitar la compresión posterior.

En la compresión temporal se compara la información entre frames consecutivos y únicamente se almacenan los detalles que varían. Si el cielo azul del ejemplo anterior fuera atravesado por un pájaro en vuelo, aplicar una compresión temporal a la secuencia implicaría describir únicamente los píxeles que variasen en cada fotograma. Se podría prescindir de la información del cielo y relacionar únicamente la del animal. No obstante es claro que la compresión temporal precisa también describir algunos fotogramas sin comprimir, en el ejemplo anterior el primero de la serie, pongamos por caso. Los fotogramas de referencia a partir de los cuáles se analizan las diferencias y se sustentan los posteriores se denominan fotogramas clave y contienen la imagen completa. Por el contrario, los fotogramas que reflejan las diferencias se denominan delta frames y sólo contienen la información de las áreas que varían respecto de las imágenes anteriores.

En general, como ya hemos apuntado anteriormente, los vídeos que presentan pocos cambios entre fotogramas se comprimen mejor y ello afecta necesariamente a la realización. Actualmente tanto la realización televisiva como la cinematográfica tienden al uso de la cámara en movimiento. No únicamente panorámicas, travellings y zooms, sino también filmaciones con "steadycam" (o su emulación como la función "steadyshot" de los equipos domésticos) e incluso filmaciones cámara en mano.

Para solucionar este tipo de problemas se utilizan algunos códecs como los Sorenson o los MPEG que tienen capacidad para tratar adecuadamente movimientos moderados de la cámara y compensar por ejemplo los cambios para lograr panorámicas sin saltos. De todos modos, el campo de estudio que relaciona los dos extremos de la cadena de trabajo, la realización y la compresión, presenta un panorama extenso y dilatado en el que hay mucho camino por recorrer.

3.3. Detección de objetos en primer plano

En procesado de imagen se entiende por detección de primer plano al conjunto de técnicas que tienen por objetivo detectar objetos en movimiento que aparecen en la secuencia de video sobre la que se trabaja.

Actualmente, el uso de sistemas inteligentes de análisis de secuencias de video está cada vez más presente en una gran variedad de sistemas: video-vigilancia [3,4], indexación de contenidos [5], cine y TV [6], compresión de video [7]. Debido a la creciente cantidad de información visual generada por las cámaras y sensores de estos sistemas, es necesario desarrollar herramientas de análisis automático, que operen en tiempo real en ciertos dominios de aplicación (por ejemplo, video-vigilancia), que permitan extraer las regiones de interés de la secuencia de video analizada. En este contexto, el primer problema es la localización de la región donde sucede algo relevante. Esta operación suele conocerse como segmentación.

El objetivo de la segmentación de objetos habitualmente consiste en diferenciar los objetos en movimiento del primer plano (frente o "foreground") de una imagen, del resto de los objetos o fondo ("background"). En el caso de una escena grabada por una cámara fija, las técnicas de segmentación más eficaces son las basadas en el modelado y posterior substracción del fondo.

La segmentación automática de objetos presenta múltiples complicaciones, siendo una de las tareas más complicadas dentro del procesado de video. Algunos de los problemas más destacados son los siguientes:

Modelo de fondo actualizable: El modelo de fondo utilizado para detectar los objetos debe poder evolucionar junto con la secuencia, con el fin de adaptarse a los cambios observados en ella, como pueden ser los cambios de iluminación o la detención de objetos de primer plano en el fondo, situación en que el objeto de primer plano pasaría a ser un objeto inmóvil de fondo.

Reducir falsas detecciones de primer plano / maximizar las detecciones correctas: Debido a que la estimación de la distribución estadística del fondo tiene como objetivo calcular la probabilidad de que una nueva muestra pertenezca a este modelo, es muy importante elegir correctamente un umbral de decisión que permita discernir entre primer plano y fondo. Pero también hay otras técnicas que consiguen el mismo objetivo, como son incluir información espacial en los modelos de fondo.

Eliminar sombras o brillos de las detecciones de primer plano: Independientemente del método de detección de primer plano, debido a que los modelos están basados en el color, es fácil detectar erróneamente sombras de objetos o brillos, pudiéndose aplicar otros algoritmos para disminuir la aparición de estas falsas detecciones.

Fondos multimodales: Son los que podemos encontrar en secuencias con objetos en movimiento lento y/o periódico (movimiento de las hojas de los árboles, movimiento ondulatorio del agua,…) y que, desde una perspectiva semántica, habitualmente se considera que pertenecen al fondo de la escena.

Camuflaje: Este efecto aparece cuando los objetos del primer plano poseen el mismo color y textura que el fondo; por este motivo, el frente se confunde o camufla como fondo.

Todo sistema de detección de objetos de primer plano se basa en el modelado del fondo y la detección de frente. El modelado de fondo consiste en la elaboración y actualización de un fondo a partir de la secuencia de video generada por una cámara y, la detección de frente es el proceso mediante el cual cada imagen de un video se compara con el modelo de fondo a fin de determinar los píxeles pertenecientes al fondo y aquellos que son primer plano o frente. Podemos ver un ejemplo en la siguiente ilustración.

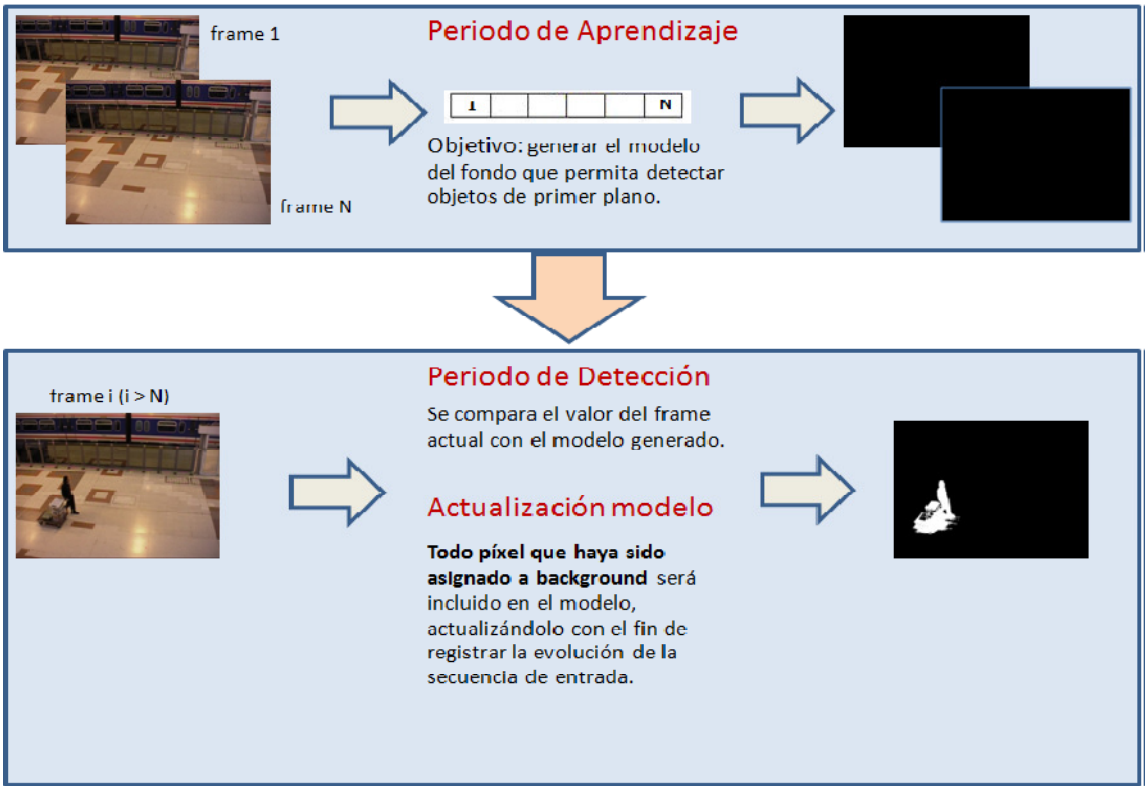


Figura 3.6: Diagrama de funcionamiento de un sistema detector de primer plano genérico.

3.3.1. Modelado del fondo

El modelado de fondo es una fase es esencial para los algoritmos de detección de objetos en movimiento. Su función es la inicialización, actualización y representación de un modelo de fondo robusto de la secuencia de video analizada. El fondo se describe mediante un modelo matemático, para cada píxel de la imagen en cada instante de tiempo.

Existen dos tipos de fondos: los fondos unimodales y los fondos multimodales. Los unimodales se dan en entornos muy controlados, en los que los valores de los pixeles del fondo no cambian a lo largo de toda la secuencia, la cámara suele ser estática y no se producen variaciones en la iluminación. Los fondos multimodales en cambio se dan en entornos menos controlados, en los cuales los píxeles del fondo pueden cambiar su valor debido a cambios de iluminación (escenas capturadas al aire libre y a distintas horas del día) o debido al movimiento de los objetos que pertenecen al fondo (como ocurre en escenas con árboles agitándose u olas de mar).

En la etapa de detección de fondo se deben tener en cuenta problemas como pueden ser los cambios de iluminación (graduales y repentinos), cambios de movimiento (oscilaciones de la cámara, movimiento de objetos con alta frecuencia,…) o cambios en

la geometría del fondo (por ejemplo coches aparcados). Para intentar solucionar estos problemas se han desarrollado una gran cantidad de algoritmos y así luego poder realizar una estimación correcta de movimiento en la secuencia analizada. Se han investigado diversas técnicas que se exponen a continuación:

- ***Métodos básicos:*** Los métodos básicos de sustracción de fondo utilizan modelos matemáticos sencillos tales como, diferencias entre imágenes, promedios, máximos y mínimos,..etc., que permiten modelar de forma simple los píxeles de la imagen de fondo.

    - *Diferencia de imágenes:* [11] Utiliza únicamente la imagen anterior como modelo de fondo. Este método solo funciona bajo determinadas condiciones de velocidad de objetos y velocidad de frames (fps). La detección es muy dependiente del umbral (Th) que seleccionemos.

    $$|frame_i - frame_{i-1}| > Th \qquad (3.5)$$

    - *Método average o median:* [12] El fondo es calculado como la media o la mediana de los previos 'n' frames en la secuencia. El problema de esta estimación es la consecuente inclusión de objetos en movimiento que pasan a formar parte del background (con el consiguiente error en posteriores detecciones). Otro problema sería el requerimiento de memoria, pues tendremos que almacenar los 'n' cuadros anteriores.

    - *Método running average o actualización progresiva de fondo:* [13] Mejora del método average evitando el uso de memoria. Sigue la siguiente fórmula:

    $$B_{i+1}(i, j) = \alpha F_i(i, j) + (1 - \alpha) B_{i+1}(i, j) \qquad (3.6)$$

    Donde típicamente el factor de actualización (α) vale 0.05.

    - *Generación de fondo basada en histogramas:* [61] Se realiza un histograma del valor de cada pixel en los últimos 'n' frames y nos fijaremos en el histograma de cada píxel para determinar si este pertenece al fondo o a un posible movimiento de un objeto, con este método podemos descartar otros factores como ruido de la imagen, cambio de luminosidad, etc.

    - *Método de selectividad:* Para cada nuevo frame, los píxeles son clasificados como fondo ("background") o primer plano ("foreground"). Si son clasificados como primer plano, entonces no se toman en cuenta para la actualización del fondo.

$$B_{i+1}(i, j) = \alpha F_i(i, j) + (1- \alpha) B_{i+1}(i, j) \text{ si } F_i(i, j) \text{ es fondo} \qquad (3.7)$$

$$B_{i+1}(i, j) = B_i(i, j) \text{ si } F_i(i, j) \text{ es fondo}$$

- **Métodos paramétricos:** Los algoritmos basados en modelos paramétricos definen modelos de fondo más complejos, que permiten cierta tolerancia al ruido y a pequeñas fluctuaciones. Describen la imagen de fondo en base a parámetros de una distribución de probabilidad estándar (usualmente Gaussiana).

    - *Método Running Gaussian Average:* [14] Se intenta ajustar una distribución gaussiana (μ, σ) sobre el histograma, así obtenemos la función de densidad de probabilidad del fondo.

    - *Método Mixture of Gaussianas:* [15] Basándonos en el método anterior, consideramos un ajuste a una mezcla de gaussianas ponderadas (μ_i, σ_i, ω_i). Con este modelo podremos hacer frente a distribuciones multimodales de fondos (backgrounds).

- **Métodos no paramétricos:** Los modelos no paramétricos también son métodos complejos en los que no se asumen distribuciones estándar de probabilidad para modelar a los píxeles de fondo, sino técnicas más generales, como pueden ser almacenamiento de los últimos valores del píxel, cálculo de rangos de valores del píxel, ajustes de funciones de predicción, etc.

    - *Método Kernel Density Estimators:* [16] La función de densidad de probabilidad (fdp) del fondo la obtenemos a través del histograma de los "n" últimos valores, cada uno alisado con un núcleo gaussiano. Si la fdp(x) > Th, entonces el píxel x es clasificado como background. Este método implica selectividad y grandes requerimientos de memoria (n * size(frame)).

    - *Métodos basados en códigos ('Codebooks'):* [17] Cada píxel de fondo Bt (x, y) puede codificarse por un conjunto de valores 'codewords' que constituirían un descriptor o 'Codebook'. Al procesar una nueva imagen de la secuencia, los valores de intensidad de los píxeles se comparan con los que forman su 'Codebook' mediante diferencias de color y luminancia, de tal forma que si existe parecido, se estiman y actualizan los códigos y, si no hay coincidencia, se inserta el nuevo código.

3.3.2. Detección de frente

La detección de primer plano también denominada detección de frente, compara la entrada con el modelo de fondo e identifica los píxeles candidatos a ser objeto en movimiento (frente o primer plano). Esta comparación se puede llevar a cabo por diferencia y la clasificación de los pixeles fijando un umbral que depende de la escena, el ruido de la cámara, y las condiciones de iluminación.

Las técnicas de detección de frente son algoritmos sencillos cuando se dispone de un modelo de fondo adecuado a la escena de análisis ya que por comparación entre la imagen actual y fondo se pueden detectar fácilmente los objetos en movimiento. Algunas de estas técnicas son las siguientes:

- **Tecnicas basadas en diferencia:** El método más básico para realizar la detección de frente *Ft* es calculando la diferencia entre dos imágenes: los píxeles cuyos valores de diferencia son altos normalmente corresponderán a objetos en movimiento.

$$(I_t(x, y) - B_t(x, y)) \leq \tau \rightarrow F_t(x, y) = 0 \qquad (3.8)$$
$$(I_t(x, y) - B_t(x, y)) > \tau \rightarrow F_t(x, y) = 1$$

  En la literatura podemos encontrar diferentes formas de diferencia:

  - *Diferencia absoluta:* [18] La diferencia absoluta entre la imagen actual y la imagen de fondo proporciona bordes gruesos que son útiles en detección de frente ya que permiten más fácilmente rellenar contornos.

  - *Diferencia relativa:* [19] Consiste en aplicar un diferencia en función de la distribución espacial de localización.

  - *Diferencia normalizada:* Otro método de detección de frente que se realiza aplicando un umbral sobre las estadísticas normalizadas.

  - *Diferencia basada en dos umbrales:* [20] Algunos autores, aplican varios umbrales a la diferencia entre la imagen y el fondo para detectar los objetos en movimiento y de esta forma, incrementar el grado de precisión en la detección.

- **Técnicas basadas en estadísticos:** Las técnicas de detección de frente basadas en estadísticos pueden utilizarse si se conoce la función densidad de ruido de la

cámara. Estos métodos, en lugar de umbralizar la diferencia de la imagen, comparan el comportamiento estadístico de un conjunto de valores de un píxel pertenecientes a una serie de imágenes de la secuencia con el píxel de la imagen actual situado en la misma posición que éstos.

Los métodos estadísticos dependen de la distribución con la que se modele el fondo, de tal forma que cada modelo de fondo utiliza una estrategia diferente para detectar los píxeles de frente. Por ejemplo:

- *Basados en una Gaussiana:* [14] La imagen actual se compara con el modelo de fondo de distribución Gaussiana (μ_t, σ_t^2) mediante la medición de la probabilidad en luminancia o color y si la probabilidad es pequeña, el píxel se clasifica como frente (Ft = 1), de lo contrario, será clasificado como fondo (Ft = 0).

$$\begin{aligned} &| I_t(x, y) - \mu_t(x, y) | \geq c\sigma_t(x, y) \rightarrow F_t(x, y) = 1 \\ &| I_t(x, y) - \mu_t(x, y) | < c\sigma_t(x, y) \rightarrow F_t(x, y) = 0 \end{aligned} \qquad (3.9)$$

- *Basados en Mezcla de Gaussianas:* [15] En una nueva imagen procesada, el valor de cada píxel se compara con las k distribuciones que modelan el píxel en una MoG (μ_t, σ_t^2) a fin de decidir si el elemento pertenece al fondo (Ft = 0), recae dentro de la Gaussiana k-ésima o al frente (Ft = 1), si la diferencia supera la varianza permitida.

$$\begin{aligned} &\textstyle\sum_{i=1}^{i=W} w_i \leq U, | \mathrm{It}(x, y) - \mu t(x, y, k)| > c\sigma t(x, y, k) \rightarrow \mathrm{Ft}(x, y) = 1 \\ &\textstyle\sum_{i=1}^{i=W} w_i > U, | \mathrm{It}(x, y) - \mu t(x, y, k) | \leq c\sigma t(x, y, k) \rightarrow \mathrm{Ft}(x, y) = 0 \end{aligned} \qquad (3.10)$$

- *Basados en la estimación de densidad de Núcleo e Histogramas:* [61] Se generan estimaciones de probabilidad para cada píxel en función de los píxeles almacenados de un número de imágenes anteriores a una procesada, de modo que, un píxel se considera primer plano (Ft = 1) si la probabilidad de pertenecer a esta estimación es inferior a un umbral.

$$\begin{aligned} &f_{x,y}(\mathit{imagen}[x, y]) > \tau \rightarrow F_t(x, y) = 0 \\ &f_{x,y}(\mathit{imagen}[x, y]) > \tau \rightarrow F_t(x, y) = 0 \end{aligned} \qquad (3.11)$$

3.3.3. Seguimiento de objetos

El paso inmediatamente posterior a la detección de los objetos en primer plano (objetos en movimiento) consiste en el seguimiento de estos objetos. El seguimiento de objetos en tiempo real es una tarea crítica en muchas aplicaciones como vídeo-seguridad, asistencia al conductor, interfaces de usuario perceptuales, compresión de video basada en objetos, etc.

En un contexto de percepción visual, el seguimiento de un objeto es, en esencia, un proceso en el que se efectúa la detección del objeto u objetos móviles y su "persecución" a través de las secuencias de imágenes del entorno adquiridas por una o varias cámaras (estáticas o móviles). El proceso de seguimiento puede implicar a cualquier objeto móvil presente en la escena, sin reconocer de qué objeto se trata o el seguimiento de uno o varios objetos específicos del cual se tiene un modelo.

Tradicionalmente existen dos aproximaciones para resolver el problema del seguimiento:

- *Las aproximaciones basadas en el análisis del contenido de los píxeles:* Estas técnicas determinan las zonas de la imagen donde se produce movimiento y realizan el seguimiento mediante un análisis del contenido de los píxeles. Ejemplos de estas dos aproximaciones son los métodos basados en el análisis de movimientos diferenciales [21] y las técnicas de flujo óptico [22].

- *Las aproximaciones basadas en modelos:* Estas técnicas implican localizar, en cada una de las imágenes que componen la secuencia, la posición del objeto móvil del que se dispone de un modelo. Los modelos empleados suelen construirse "a mano", inducirse a partir de una secuencia de ejemplos o adquirirse dinámicamente a partir del objeto móvil. Los métodos basados en modelo incluyen: métodos correlacionales [23], métodos basados en correspondencia [24], métodos de filtrado [25], métodos basados en contornos deformables activos [26] y métodos basados en filtrado predictivo [27].

3.4. Detección y reconocimiento de acciones

El concepto de análisis de movimiento se puede definir, entendiendo análisis, como la separación de las partes de un todo hasta conocer sus principios y/o elementos, y movimiento, como "cambio continuo en la posición de un objeto". Por lo tanto el análisis de movimiento se puede definir como la búsqueda de los elementos y principios que permiten los continuos cambios de posición del cuerpo humano.

El análisis de movimiento no es un tema actual, ya en la Antigua Grecia, Aristóteles (348-322 AC) veía el cuerpo de los animales como sistemas mecánicos, lo que expuso en su libro "De Motu Animalium". Sorprendente resulta saber que él ya cuestionaba las diferencias fisiológicas entre la acción imaginada y la desarrollada. Mil años después, Leonardo Da Vinci (1452-1519) describiría la mecánica de la bipedestación, marcha en descenso y ascenso, levantamiento desde sentado y salto, en sus dibujos anatómicos. Cien años después, Galileo Galilei (1564-1643) haría los primeros intentos por analizar matemáticamente la función fisiológica y Borelli (1608-1679) apoyado en dicho trabajo dibujó fuera de las articulaciones las fuerzas requeridas para el equilibrio del cuerpo. Algo más tarde Newton publicó las "Leyes del Movimiento" y determinó la posición del centro de gravedad humano. Muybridge (1830-1904) es el primero en disecar el movimiento humano y animal, a través de una secuencia fotográfica, luego Marey (1830-1904) utilizaría científicamente esta técnica para correlacionarla con la fuerza de reacción del suelo, hito que marca el inicio del Análisis de Movimiento Moderno.

Actualmente los videos se crean y se propagan muy fácilmente por los medios de comunicación y están al servicio del entretenimiento, de la comunicación y otros fines. La demanda asociada de extraer las grandes colecciones de datos de vídeo realistas motiva a ir más lejos en la investigación para el entendimiento automático de vídeo. Es de gran interés práctico y científico comprender los movimientos articulados cuerpo, especialmente los del cuerpo humano. Desde la visión de la computación, una cuestión interesante es como representar y reconocer los diferentes tipos de movimientos humanos con modelos eficaces.

Imagínese un vídeo tomado una playa soleada, donde hay gente jugando al voleibol de playa, unos hacen surf, y otros dan un paseo a lo largo de la playa. ¿Automáticamente puede un ordenador decirnos qué pasa en la escena? ¿Puede identificar diferentes acciones humanas? Actualmente se está estudiando el problema de detección y la clasificación de la acción humana en secuencias de vídeo.

La tarea de clasificación automática y localización de acciones humanas en secuencias de vídeo es sumamente interesante para una gran variedad de aplicaciones: descubrimiento de actividades relevantes en vídeos de vigilancia, resumiendo e indexando secuencias de vídeo, organizando una biblioteca digital de vídeo según las acciones relevantes, etc. Sigue siendo, sin embargo, un problema difícil para los ordenadores lograr el reconocimiento robusto de acción debido al fondo desordenado, el movimiento de cámara, la oclusión, cambios de punto de vista, y las variaciones geométricas y fotométricas de objetos.

En este apartado vamos a comentar algunos de los sistemas o algoritmos, que podemos encontrar en la literatura, que trabajan en sobre el problema de la detección y reconocimiento de las acciones humanas.

3.4.1. "A Hierarchical Model of Shape and Appearance for Human Action Classification"

En el documento "A Hierarchical Model of Shape and Appearance for Human Action Classification" [28] se presenta un modelo para caracterizar las acciones humanas, en particular, se propone un modelo que tiene en cuenta tanto rasgos estáticos como dinámicos del movimiento humano [29,30]. Una secuencia de video se representa como una colección de características espaciales y espacio-temporales mediante la extracción de puntos estáticos y dinámicos de interés.

La utilización de las mejores características para describir la postura y el movimiento ha sido un tema ampliamente estudiado en los últimos años. Por lo general, hay tres tipos de rasgos más utilizados: rasgos estáticos basados en bordes y la forma de los miembros [31], rasgos dinámicos basados en flujos ópticos [32], y los rasgos espacio-temporales que caracterizan un volumen espacio-temporal de los datos [29,33]. Las características espacio-temporales han demostrado ser especialmente prometedoras en la marcha para la comprensión del movimiento debido a su rico poder descriptivo. Pero por otro lado, si sólo dependemos de estas características significa que únicamente se podría caracterizar los movimientos en los vídeos. Sin embargo si nos basamos en la experiencia de la vida cotidiana, nos damos cuenta que los seres humanos pueden reconocer perfectamente un movimiento basándose en un solo gesto. Por lo tanto en este trabajo se propone el uso híbrido de rasgos de forma estáticos así como rasgos espaciales temporales.

Por otro lado para objetos estructurados como el cuerpo humano, es importante modelar la relación geométrica mutua entre las diferentes partes. Los modelos de constelación ofrecen una solución de este tipo [30]. Lamentablemente, debido a la complejidad computacional del modelo, sólo se puede utilizar un número muy reducido de características. Otro enfoque es perder toda la información geométrica y considerar los modelos de "bag of words". Estos han resultado ser sumamente eficientes y eficaces en la clasificación de objetos y del movimiento humano. En el documento se propone un método de explotar a ambos, el poder geométrico del modelo de constelación así como la riqueza del modelo de "bag of words". Reconociendo el límite computacional de tener un número muy pequeño de partes totalmente conectadas en el modelo de constelación. Pero en vez de aplicarse esto directamente en los rasgos de la imagen, se adjunta el modelo de "bag of words" a cada parte del modelo de constelación. La representación general incorpora un modelo jerárquico que combina un modelo de constelación de pocas partes con modelos de "bag of words" con un número grande y flexible de rasgos.

En la versión más simple, el modelo costa de dos niveles jerárquicos. La capa superior esta cercana en forma a las condiciones del modelo de constelación, se compone de un conjunto de P partes cuya posición está representada por una densidad Gaussiana de sus posiciones relativas. Cada una de las partes P_p (p = 1... P) está conectada a N_p rasgos de imagen en el nivel inferior, y a su vez está asociado a las distribuciones de aparición y ubicación relativa de las características que se le asignen. En otras palabras, la capa superior es una constelación de partes, y cada una de estas partes se asocia a una "bolsa de características" en la capa inferior. Debido a sus limitaciones geométricas, este modelo es conveniente para capturar configuraciones de cuerpo similares o posturas.

Tras la observación de que las acciones humanas son el resultado de secuencias de posturas, que surgen de unos pocos conjuntos de configuraciones de cuerpo similares, se llega a la conclusión de que una única acción está mejor representada como una distribución multimodal de forma y apariencia. Para tener en cuenta esta multimodalidad, se usa una mezcla de modelos jerárquicos, donde cada uno de los componentes corresponde a una serie de poses que se agrupan según su semejanza.

Considerando un nuevo frame de vídeo y los modelos aprendidos para cada clase de acción, la tarea es clasificar los nuevos datos que pertenecen a uno de los modelos de acción. Suponga que se tiene C números de clases. Se calcula la probabilidad de observar los datos de imagen dados que han sido generados de cada clase C. Esto produce un vector de características C-dimensional de la entrada en el espacio modelo. Se calcula este vector de rasgos para cada ejemplo en un conjunto de validación, y se usa para entrenar un clasificador discriminatorio. Por lo tanto, una decisión de

clasificación se hace primero calculando la probabilidad de la entrada según cada uno de los C modelos de acción, y luego categorizando este vector de características C-dimensional usando el clasificador discriminativo.

Además, las decisiones pueden ser hechas sobre una gama de frames de vídeo adoptando una estrategia de bolso-de-frames. Primero, cada frame es clasificado por separado, y se le asigna un voto a favor de una clase de acción. La secuencia completa de vídeo es clasificada por ser la categoría que reune la mayoría de los votos.

En resumen el objetivo de este enfoque es conseguir un modelo jerárquico que puede caracterizarse como una constelación de bolsas de características y que es capaz de combinar características espaciales y espacial-temporales cuya finalidad es aprender y clasificar las diferentes categorías de movimiento humano.

3.4.2. "Unsupervised Learning of Human Action Categories Using Spatial-TemporalWords"

En el trabajo "Unsupervised Learning of Human Action Categories Using Spatial-TemporalWords" [34] se quiere diseñar un algoritmo que permite a la computadora aprender modelos de las acciones humanas. Entonces, dado un nuevo vídeo, el algoritmo debe ser capaz de decidir qué acción humana está presente en la secuencia. Además, se quiere buscar el medio de proporcionar una indicación preliminar de donde está (en el espacio y el tiempo) siendo realizada la acción.

El fondo desordenado, el movimiento de cámara, la oclusión, los cambios de punto de vista, y las variaciones geométricas y fotométricas de objetos son problemas que causan que pocos algoritmos de visión puedan identificar, clasificar y localizar los movimientos de forma adecuada. Además, el desafío es aún mayor cuando hay múltiples actividades en una secuencia compleja de vídeo. En este trabajo se presenta un algoritmo que está orientado a explicar estos escenarios.

Se propone en este artículo un acercamiento de un modelo gráfico generativo para aprender y reconocer acciones humanas en vídeo, aprovechando la representación robusta de los escasos puntos de interés espacio-temporales y un enfoque de estudio no supervisado. El estudio no supervisado se logra obteniendo parámetros del modelo de acción provenientes de secuencias de vídeo sin segmentar ni etiquetar, que contienen un número conocido de clases de acción humanas. Se emplea un entorno de estudio sin supervisión porque esto abre la posibilidad de aprovechar la cantidad creciente de datos disponibles de vídeo, sin el costo de la detallada anotación humana. Con este fin, un

enfoque generativo proporciona los medios para aprender los modelos de una manera no supervisada, en contraposición a los modelos de discriminación que, en general, requieren datos detallados etiquetados.

Este método está motivado por el éxito de la detección/clasificación de objeto o la clasificación de escena de imágenes estáticas sin etiquetar, usando modelos de tema latentes [35]. Una consideración clave en estos trabajos es conocida como la representación " bag of words", donde la disposición geométrica entre características visuales es ignorada. Esto comúnmente se implementa como un histograma del número de casos de patrones visuales particulares en una imagen dada.

Dado un conjunto de secuencias de vídeo sin etiquetar este método pretende descubrir un conjunto de clases de ellos. Cada una de estas clases correspondería a una categoría de acción, tal que se puede construir modelos para cada clase. Además, pretende entender los vídeos que están compuestos por una mezcla de categorías de acción, para manejar el caso de múltiples movimientos. Esto se parece al problema del descubrimiento de tema automático en el análisis de texto. En este caso, se quiere analizar secuencias de vídeo en vez de documentos de texto; las secuencias de vídeo son resumidas como un conjunto de palabras espacio-temporales en vez de palabras de texto; se procura descubrir categorías de acción en vez de temas de texto; y los vídeos son explicados como una mezcla de acciones en vez de documentos de texto como una mezcla de temas.

Con este trabajo, se investigan dos modelos que se propusieron en la literatura de análisis de texto para dirigir el problema de descubrimiento de tema latente: probabilistic Latent Semantic nalysis (pLSA) by Hofmann (1999) [36] and Latent Dirichlet Allocation (LDA) by Blei et al. (2003) [37]. En este trabajo, se investiga la conveniencia de ambos modelos para el análisis de vídeo explorando las ventajas de la poderosa representación y la gran flexibilidad de estos modelos gráficos generativos. Así pues, estos modelos proporcionan un marco de aprendizaje sin supervisión, que permite descubrir automáticamente agrupaciones semánticas en el entrenamiento de datos. También, a diferencia de métodos discriminatorios como Support Vector Machines, pLSA y LDA permiten al algoritmo realizar el razonamiento significativo sobre los datos más allá de la clasificación, por ejemplo la localización de asunto. Además, tal localización puede ser realizada sin la necesidad de explorar miles o millones de ventanas por imagen. Estos modelos, sin embargo, no proporcionan invariancias de escala espacial ni temporal. Así, estos sólo pueden trabajar dentro de un pequeño margen de las escalas que ha sido observado en el entrenamiento.

En este sistema se representa cada secuencia de vídeo como una colección de palabras espacio-temporales extrayendo puntos de interés de espacio-tiempo. Se usa el método de filtro lineal separable, ya que generalmente produce un alto número de detecciones. Como podemos ver en la figura 3.7 pequeños pedazos de vídeo son extraídos de cada punto interés y constituyen la información local que se utiliza para aprender y reconocer categorías de acción humanas. Empleando rasgos locales, se intenta acentuar la importancia del corto rango de patrones espacio-temporales. Los patrones locales observados son bastante discriminatorios a través de clases de acción humanas, y proporcionan un espacio característico razonable que permite construir buenos modelos de acción humana.

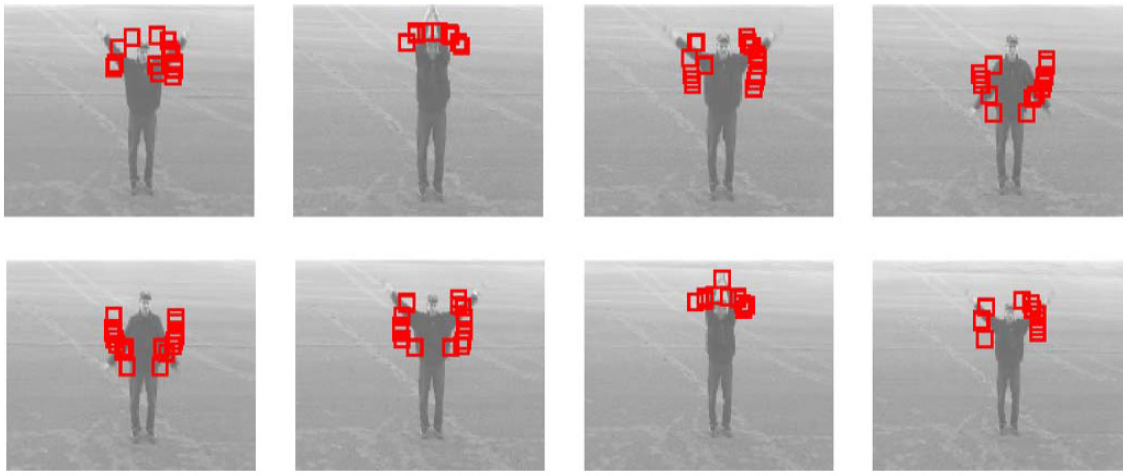


Figura 3.7: Muestra la detección de los puntos de interés usando el método de filtro lineal separable.

Para obtener un descriptor para cada cubo espacial temporal, se calculan sus gradientes de brillo sobre 'x', y 'y' direcciones de 't'. El cubo espacial temporal entonces es alisado en escalas diferentes antes del cálculo de los gradientes de imagen. Los gradientes calculados se concatenan para formar un vector.

El descriptor que se utiliza es un descriptor muy simple basado en gradientes de imagen, tal descriptor no proporciona invariancia de escala, ni en el espacio, ni dominio de tiempo, tampoco captura el movimiento relativo de la cámara. Sin embargo, los descriptores más complejos que incluyen pequeñas invariancias a la escala espacial y a la velocidad, así como invariancias a pequeños movimientos de cámara, tienen una mayor complejidad computacional. En este acercamiento se confía en el codebook para manejar cambios de escala y movimientos de cámara. Mientras los rasgos locales recién observados no contienen el modelo de cambio de escala y ni el movimiento de cámara que es sumamente diferente de los observados en los datos utilizados para formar el codebook, se espera que rasgos locales similares sean asignados a miembros constantes en el codebook.

Los modelos tema latente pLSA y LDA se basan en la existencia de un vocabulario finito de las palabras espacio-temporales de tamaño V. Para aprender el vocabulario de palabras espacio-temporales, consideramos el juego de descriptores correspondiente a todos los puntos de interés espacio-temporales descubiertos en los datos que se entrenan. Este vocabulario es construido por agrupación utilizando el algoritmo K-means y la distancia euclídea como métrica de agrupación. El centro resultante de cada grupo se define como una palabra espacio-temporal. Así, cada punto de interés descubierto puede ser asignado a un único grupo.

Una característica importante de los modelos pLSA y LDA es que están basados en la suposición de bolso de de palabras, es decir el orden de las palabras en un documento de texto puede ser descuidada. En el contexto de clasificación de acción humana, la suposición de bolso de palabras se traduce en una representación de vídeo que ignora el convenio posicional, en el espacio y el tiempo, de los puntos de interés espaciales temporales. Tal suposición trae las ventajas de usar una representación simple que hace el aprendizaje eficaz. La carencia de información espacial proporciona poca información sobre el cuerpo humano, mientras la carencia información temporal a largo plazo no permite modelar las acciones más complejas que no están constituidas por patrones simples repetitivos.

Cuando la secuencia de pruebas es considerablemente larga, la secuencia se divide en subsecuencias utilizando una ventana temporal que se desliza. Tales subsecuencias son tratadas por separado y obtenemos decisiones de clasificación para cada uno de ellas. Esto es necesario debido a la naturaleza de la representación: la carencia de ordenamiento temporal relativo de características en nuestra representación de " bag of words" no proporciona el medio para asignar etiquetas en los casos de tiempo diferentes dentro de un vídeo; en cambio, el análisis es hecho para la secuencia completa. Así, dividiendo el vídeo largo original en subsecuencias, el método puede asignar etiquetas a cada subsecuencia dentro de la secuencia larga.

Las contribuciones de este trabajo son dobles. Primero, se propone un acercamiento de estudio no supervisado para acciones humanas que usan una representación de bolso de palabras. Se aplican dos modelos de asunto latentes, pLSA y LDA, al problema de aprender y reconocer categorías de acción humanas, adoptando la representación de "bag of spatial-temporal words " para secuencias de vídeo. Segundo, este método puede localizar y clasificar múltiples acciones en un solo vídeo. Además de la tarea de clasificación, este acercamiento también puede localizar acciones diferentes simultáneamente en una secuencia nueva y compleja de vídeo. Esto incluye los casos donde varias personas realizan acciones distintas al mismo tiempo, y también

situaciones donde una persona sola realiza acciones distintas durante el tiempo que dura una secuencia de vídeo.

3.4.3. "Actions in Context"

El artículo "Actions in Context" [38] nos da un enfoque para entender las secuencias de vídeo basándose en el estudio conjunto de las acciones, los escenarios y los objetos, beneficiándose así de las limitaciones contextuales mutuas que hay entre ellos.

Este trabajo se construye sobre la susodicha intuición y explota relaciones de concurrencia entre acciones y escenas en el vídeo. A partir de un determinado conjunto de clases de acciones, se pretende descubrir automáticamente las clases de escena correlacionadas y usar esta correlación para mejorar el reconocimiento de acción. Ya que algunas acciones son relativamente independientes de la escena (p.ej. "la risa"), no se espera que el contexto sea igualmente importante para todas las acciones. El contexto de escena, sin embargo, esta correlacionado con muchas clases de acción. Es por lo tanto esencial para el reconocimiento de acción en general.

En este acercamiento se exploran escenas y acciones en vídeos genéricos y realistas. Se evitan situaciones específicas tales como vigilancia o deportes, y se considera un conjunto grande y diverso de muestras de vídeo de películas, se usan guiones de película para la anotación automática de vídeo y se aplica la extracción de texto para descubrir las clases de escena que concurren con las acciones dadas. Se utiliza la alineación de guion-a-vídeo y la búsqueda de texto para recuperar automáticamente las muestras de vídeo y las etiquetas correspondientes para las escenas y acciones en películas.

Los guiones son documentos de texto disponibles públicamente para la mayoría de las películas populares, contienen títulos de escena, diálogos y descripciones de escena, pero por lo general no ofrecen la sincronización de tiempo con el vídeo. Este problema se aborda utilizando los subtítulos que si están sincronizados con el vídeo. Así para conseguir la alineación guion-a-video se utiliza la información de tiempo de los subtítulos para poder estimar la localización temporal de los títulos de escena y las descripciones de escena.

Mejorar la clasificación de acciones explotando relaciones que ocurren entre acciones y su contexto de escena es la finalidad principal de este trabajo. Esto se consigue para un conjunto dado de clases de acción mediante (i) la identificación de tipos de escena relevantes (ii) y la estimación de la relación de concurrencia con las acciones. En este

punto es donde se recurre a los guiones de cine, anteriormente sincronizados con el vídeo, ya que usando los títulos de escena y las descripciones cortas de escena se obtiene la información requerida sobre los tipos de escena y su concurrencia con las acciones. Una vez tenemos esta información se pueden recuperar automáticamente las muestras de vídeo con las etiquetas correspondientes para las escenas y acciones en películas.

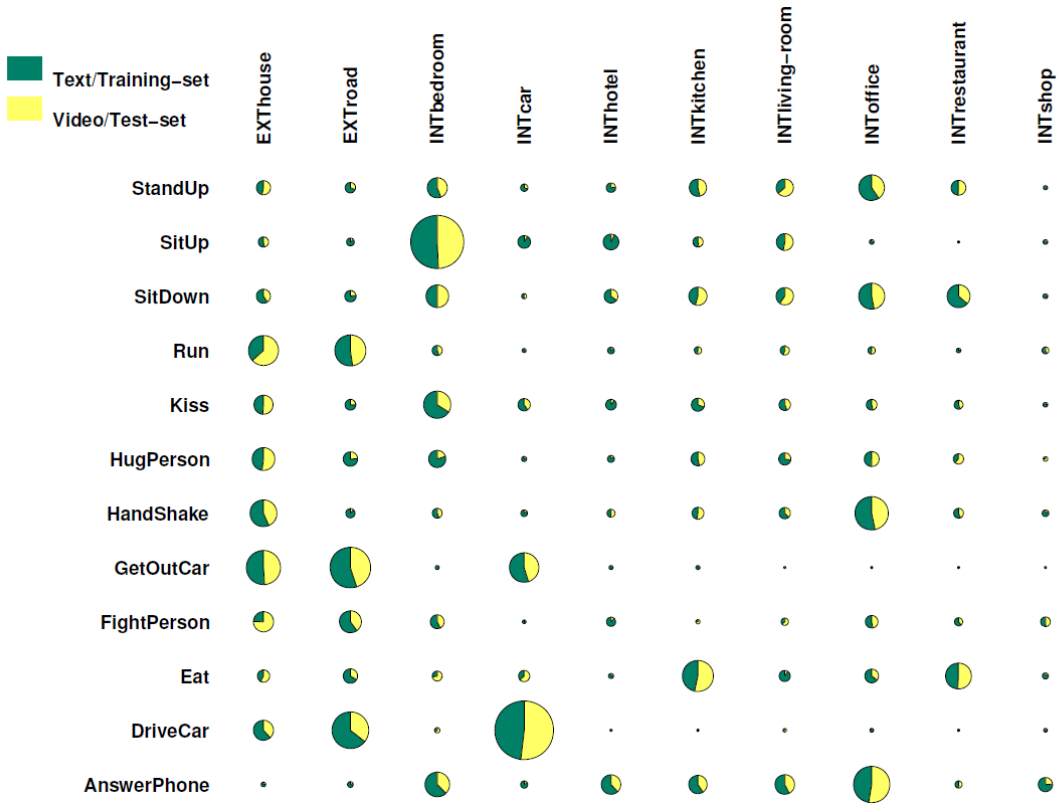


Figura 3.8: Ilustración sobre la concurrencia entre las clases de escena y las acciones.

El tamaño de los círculos de la figura 3.8 corresponde a las probabilidades estimadas de que una clase de acción se dé en una determinada escena y coincide así con las expectativas intuitivas. Por ejemplo, correr sobre todo ocurre en escenas exteriores, mientras que comer ocurre en escenas de restaurante y en la cocina. El color verde de los círculos corresponde a las concurrencias calculadas automáticamente de entrenar guiones y el color amarillo corresponde a las concurrencias manualmente evaluadas sobre las películas de prueba, como se puede observar el reparto dentro de los círculos es equitativo por lo tanto se valida la coherencia entre las relaciones extraídas del texto y lo que realmente se produce en los vídeos.

Considerando las muestras de vídeo recuperadas automáticamente con etiquetas posiblemente ruidosas, se usa la representación de bolsa de características y SVM para aprender modelos visuales separados para la clasificación de escena y la acción.

La representación de bolsa de características ve los vídeos como volúmenes espacio-temporales. Considerando una muestra de vídeo, las principales regiones locales son extraídas usando un detector de punto de interés. Después, el contenido de vídeo en cada una de las regiones detectadas es descrito con descriptores locales. Finalmente, las distribuciones desordenadas de rasgos son calculadas para toda la muestra de vídeo. Para construir la base de referencia se extrae movimiento basado en las características espacio-tiempo. Esta representación se centra en las acciones humanas consideradas como patrones de movimiento. Las regiones salientes descubiertas corresponden al movimiento y los rasgos están basados en la dinámica de gradiente y el flujo óptico. Este enfoque básico ha demostrado el éxito en el reconocimiento de la acción, pero por sí solo no es suficiente para la clasificación de la escena. Para tener modelos de escenas fiables, es necesario incluir el aspecto estático, además de los patrones de movimiento. Por lo tanto el vídeo también es visto como una colección de frames. Esto permite emplear los componentes de bolsas de características desarrollados para la escena estática y el reconocimiento de objeto y tener una representación híbrida en un marco de trabajo unificado.

Usamos el Support Vector Classifier [39] (SVM) para aprender el contexto de escena y acciones. La función de decisión de un clasificador C-SVM binario tiene la forma siguiente:

$$g(x) = \sum_i \alpha_i y_i K(x_i, x) - b \tag{3.12}$$

El kernel más simple posible es un kernel lineal. La función de decisión entonces puede ser vuelta a escribir como una suma ponderada de componentes de muestra.

$$K(x_i, x_j) = x_i x_j, \qquad g(x) = wx - b \tag{3.13}$$

Como la principal contribución de este documento se demuestra que podemos usar la información automáticamente extraída del contexto basada en guiones de vídeo y mejorar el reconocimiento automático de las relaciones contextuales en ambos tanto en (i) el reconocimiento de las acciones en el contexto de las escenas, así como (ii) el reconocimiento de escenas en el contexto de las acciones.

3.4.4. Otros métodos de clasificación de la acción humana

Una cantidad considerable de trabajos anteriores ha abordado la cuestión de la categorización de la acción humana y el análisis del movimiento. Una línea de trabajo está basada en el cómputo de la correlación entre los volúmenes de datos de vídeo. Efros et al. (2003) [40] realizan el reconocimiento de acción correlacionando medidas de flujo ópticas de vídeos de baja resolución. Su método requiere, en primer lugar la segmentación y la estabilización de cada figura humana en la secuencia, así como una mayor intervención humana para anotar las acciones en cada volumen resultante espacio-temporal. Shechtman y Irani (2005) [41] proponen una correlación basada en el comportamiento para calcular la semejanza entre los volúmenes de espacio-tiempo que permite encontrar comportamientos y acciones similares. En cada pixel, los gradientes de espacio-tiempo del pedazo correspondiente de vídeo deben ser calculados y resumidos en una matriz. Los valores propios de las matrices resultantes son usados para calcular la semejanza entre dos pedazos espacio-temporales. Por lo tanto, este método requiere el cómputo significativo debido al procedimiento de correlación entre cada pedazo de la secuencia de pruebas y la base de datos de vídeo.

Otro acercamiento popular es primero rastrear partes de cuerpo y luego usar las trayectorias de movimiento obtenidas para realizar el reconocimiento de acción. Esto se realiza con mucha supervisión humana y la robustez del algoritmo es sumamente dependiente del sistema de seguimiento. En el acercamiento de reconocimiento de acción de Ramanan y Forsyth (2004) [42] en primer lugar se hace un seguimiento de los seres humanos en las secuencias usando un procedimiento de estructura pictórico. Entonces las configuraciones 3D del cuerpo son estimadas y comparadas en una biblioteca anotada de movimiento 3D. El algoritmo permite la asignación de etiquetas compuestas a las secuencias de pruebas; sin embargo, esto depende en gran medida el resultado del seguimiento, y la estimación de la postura 3D puede introducir errores significativos debido a ambigüedades difíciles de solucionar. En Yilmaz y Cha (2005) [43], el etiquetaje humano de puntos de referencia del cuerpo humano primero es realizado en cada frame en secuencias de múltiples cámaras en movimiento. Entonces las acciones son comparadas usando su correspondientes trayectorias 4D (x, y, z, t). Así, su acercamiento puede ser aplicado al reconocimiento y la recuperación de acción, con el coste de una cantidad significativa de anotación humana.

También, los investigadores han considerado el análisis de acciones humanas mirando las secuencias de vídeo como volúmenes de intensidad de espacio-tiempo. Blank et al. (2005)[44] representan las acciones como formas de espacio-tiempo y extraen las características de espacio-tiempo para el reconocimiento de acción, como la

prominencia de espacio-tiempo local, la acción dinámica, la forma y la orientación de las estructuras. Asimismo este acercamiento se basa en la restricción de los fondos estáticos que les permite segmentar el primer plano usando la substracción de fondo.

Algunos investigadores también han explorado métodos sin supervisión para el análisis de movimiento. Zhong et al. (2004) [45] han propuesto un acercamiento no supervisado para descubrir la actividad insólita en secuencias de vídeo. Usando una representación global basada en un vector descriptor simple por cada frame, el método agrupa segmentos de vídeo e identifica grupos espacialmente aislados como la actividad insólita. Por lo tanto, las actividades insólitas deben ser observadas durante el entrenamiento.

Otros acercamientos usan una representación de vídeo basada en puntos de interés espacio-temporales. Ke et al. (2005)[46] aplican rasgos espaciales temporales volumétricos que de manera eficiente exploran secuencias de vídeo en el espacio y el tiempo Su acercamiento descubre puntos de interés sobre vectores de movimiento, lo que requiere una estimación densa del flujo óptico. Además, el método requiere calcular un número significativo de características que son del orden de un millón, aún después de la discretización y el muestreo del espacio característico. Los puntos de interés detectados entonces son empleados como características para realizar la clasificación de la acción humana con un clasificador discriminatorio de cascada, que requiere ejemplos positivos y negativos anotados. En (Schuldt et al. 2004 [47]; Dollár et al. 2005 [48]; Oikonomopoulos et al. 2006 [49]; Ke et al. 2005 [46]), los puntos de interés de espacio-tiempo son combinados con clasificadores discriminatorios para aprender y reconocer acciones humanas. Por lo tanto, pedazos locales de espacio-tiempo han demostrado ser útiles para proporcionar significado semántico de acontecimientos de vídeo proporcionando una representación compacta y abstracta de modelo.

Finalmente, notamos el éxito de enfoques generativos basados en modelos de temas latentes para el reconocimiento de escena y objeto. Fei-Fei y Perona (2005) [35] introducen el uso de modelos de asunto latentes a tareas de visión de ordenador, en el ámbito de la categorización natural escena. Sus modelos son inspirados por el modelo de LDA (Blei et al. 2003 [37]), y puede aprender distribuciones de asunto intermedias sin supervisión. También, Sivic et al. (2005) [50] realizan el aprendizaje y el reconocimiento sin supervisión de clases de objeto aplicando un modelo de pLSA con la representación visual de bolso de palabras. El acercamiento permite aprender clases de objeto de imágenes sin la etiqueta y sin desorden del fondo.

4. Diseño

4.1. Introducción

El reconocimiento de las actividades dinámicas es necesario en varios sistemas de análisis de video, incluyendo sistemas de seguridad, espacios inteligentes, video indexado basado en acciones, browsing, clustering y segmentación. Algunos de los trabajos anteriores se centran en reconocer una serie de acciones predefinidas, o entornos restringidos de imágenes. Estos métodos proponen aproximaciones para capturar las características importantes de las acciones mediante modelos paramétricos especializados que por lo general dan lugar al reconocimiento de alta calidad de las acciones estudiadas, pero en su defecto su construcción, generalmente requiere una fase de aprendizaje muy amplia, donde se proporcionan muchos ejemplos de la acción estudiada. Sin embargo, este trabajo no sólo se restringe al reconocimiento de las acciones cuidadosamente estudiadas, sino que tratamos con datos generales de vídeo en los que a menudo no hay ningún conocimiento previo sobre los tipos de acciones en la secuencia de vídeo, su grado temporal y espacial, o su naturaleza (periódica/no periódica).

El principal objetivo en este proyecto es reconocer una serie de acciones humanas en vídeo basándose en los datos de un conjunto de actividades ejemplo guardadas en una base de datos. El diseño del algoritmo permite a la computadora, en una primera etapa, aprender modelos de las acciones humanas, y más adelante dado un nuevo vídeo, el algoritmo debe ser capaz de decidir si las acciones humanas aprendidas anteriormente están o no presentes en la secuencia.

Esto se consigue mediante una medida de semejanza, basada sólo en el comportamiento, entre las secuencias de vídeo. Es decir, dicha medida conserva las variaciones temporales, mientras es insensible a cambios de aspecto como la variación de la ropa, condiciones de iluminación, etc. Esta medida es no paramétrica pudiendo manejar así una amplia gama de comportamientos dinámicos. Por lo tanto no puede ser óptimo para una acción específica, pero permite el análisis general basado en el comportamiento de información de vídeo que contiene tipos de acciones desconocidos.

Para realizar este proyecto se ha tomado como punto de partida el trabajo de Zelnik-Manor e Irani[1], descrito en el documento "Statistical Analysis of Dynamic Actions", cuyo algoritmo se ha adaptado y mejorado en base a los objetivos de este proyecto.
Algunas posibles aplicaciones para este sistema podrían ser:

- Detección de actividades inusuales en un video de seguridad en un lugar público.

- Indicar un punto interesante de vídeo que contiene una acción de interés, y requerir al sistema avanzar rápido al siguiente fragmento en el que ocurran acciones similares.

- Segmentación temporal basada en comportamientos de secuencias largas de vídeo. Dada una secuencia larga de video que contiene variedad de acciones nos interesaría detectar los puntos de comienzo y final de las acciones, sin requerir cualquier conocimiento a priori de los tipos de acciones o sus grados temporales.

4.1.1. Definición de acción

Las acciones son amplios términos temporales, que normalmente se extienden sobre decenas y centenas de frames. Polana y Nelson[60] separaron la clase objetos temporales en tres grupos y sugirieron distintas aproximaciones para el modelado y el reconocimiento de cada uno:

- Texturas temporales que tienen indefinidas las áreas espacial y temporal.

- Actividades que son temporalmente periódicas pero restringidas espacialmente.

- Eventos de movimiento que son acciones aisladas que no se repiten ni en espacio ni en tiempo.

Nuestros experimentos están enfocados principalmente en los dos últimos puntos, debido a su gran número de aplicaciones y su interés científico.

Inicialmente haremos una serie de observaciones que afectarán al diseño de nuestro esquema de modelado de acciones. Primero, observamos que una representación que puede manejar los tres tipos de acciones/comportamientos ha de ser general, por ejemplo no se pueden hacer severas suposiciones, tales como estacionalidad (que es común en modelado de texturas temporales) o periodicidad (que es común en el

modelado de actividades repetitivas). Segundo, la representación basada en comportamientos tiene que depender de características basadas en el movimiento, las cuales son invariantes frente a cambios en apariencia tales como aquellos causados por variaciones en el vestuario, en las condiciones de luz, en el fondo, etc. Por último hemos de diferenciar que acciones están caracterizadas por múltiples escalas temporales. Por ejemplo, en una secuencia de personas caminando, la resolución más alta conseguirá capturar el movimiento de los miembros, mientras que las resoluciones temporales bajas capturarán principalmente el movimiento global del cuerpo entero.

Deberíamos tener en cuenta que aunque las acciones son capturadas en múltiples escalas temporales, una acción específica está siempre realizada con aproximadamente la misma velocidad, y por esto, es capturada en la misma escala temporal. Por ejemplo, un único paso de una persona andando visto por dos cámaras de video diferentes con la misma tasa de frame, se extenderá sobre el mismo número de frames en ambas secuencias, sin tener en cuenta los parámetros internos y externos de la cámara. Similarmente, un único paso de dos personas diferentes se extenderán también sobre el mismo número de frames. Ésta observación implica que ante la diferenciación de dos acciones, no es necesario comparar a través de escalas temporales, aunque es preferible. Esto facilita el diseño de una medida de similitud secuencia a secuencia.

Un escenario que no cumple esta suposición es en el que una misma acción es realizada con velocidades significativamente diferentes. En este caso, comparar escalas temporales correspondientes debe ser insuficiente para detectar la similitud entre las acciones. Sin embargo, en este caso, no se puede afirmar directamente si dos videos han de ser considerados como capturas de la misma acción o no. Por ejemplo, es lo mismo andar a paso ligero que correr, o es lo mismo un baile rápido que uno lento.

4.2. Software utilizado

Para completar este proyecto se ha pasado por dos etapas diferenciadas a la hora de la implementación del algoritmo. Primero se implementó una versión inicial del proyecto en lenguaje MATLAB, debido a que ofrece bastantes ventajas para trabajar con matrices, y por lo tanto con imágenes y vídeo. En esta etapa se experimentó con diferentes soluciones posibles para lograr los objetivos propuestos y se escogieron los métodos que más convenían para la consecución de este trabajo.

En la segunda etapa, basándonos en los experimentos de la etapa anterior, el algoritmo definitivo se ha implementado en el lenguaje de programación C++ sobre una

plataforma PC. Ya que este sistema trabaja sobre imágenes y vídeos nos hemos ayudado principalmente de las librerías proporcionadas por el paquete OpenCV.

4.2.1. MATLAB

MATLAB (abreviatura de MATrix LABoratory, "laboratorio de matrices") es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Apple Mac OS X.

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con las cajas de herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets).

Es un software muy usado en universidades y centros de investigación y desarrollo, ya que se puede considerar como uno de los softwares más utilizados en ingeniería.

4.2.2. OpenCV

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

Open CV es multiplataforma, Existiendo versiones para Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica.

4.3. Estructura general

En esta sección se describe la estructura del sistema global (Figura 4.1) y se hará una introducción de las técnicas usadas en el desarrollo. El sistema general se divide en tres fases principales. Estas tres fases se corresponden con la estructura de los sistemas automáticos de reconocimiento de la cual ya se ha hablado antes en el apartado de fundamentos de la biometría. A estas tres fases principales se puede añadir una nueva fase que sería el procesado de la señal la cual, a pesar de no ser esencial, nos va a permitir entender y manejar de una forma más eficiente los datos de entrada y va a ayudar a mejorar el sistema de reconocimiento. De esta manera el sistema queda dividido en las siguientes etapas:

4.3.1. Captura de la secuencia de video, en esta fase se puede añadir el procesado de la señal.

4.3.2. Extracción de características.

4.3.3. Comparador y decisor.

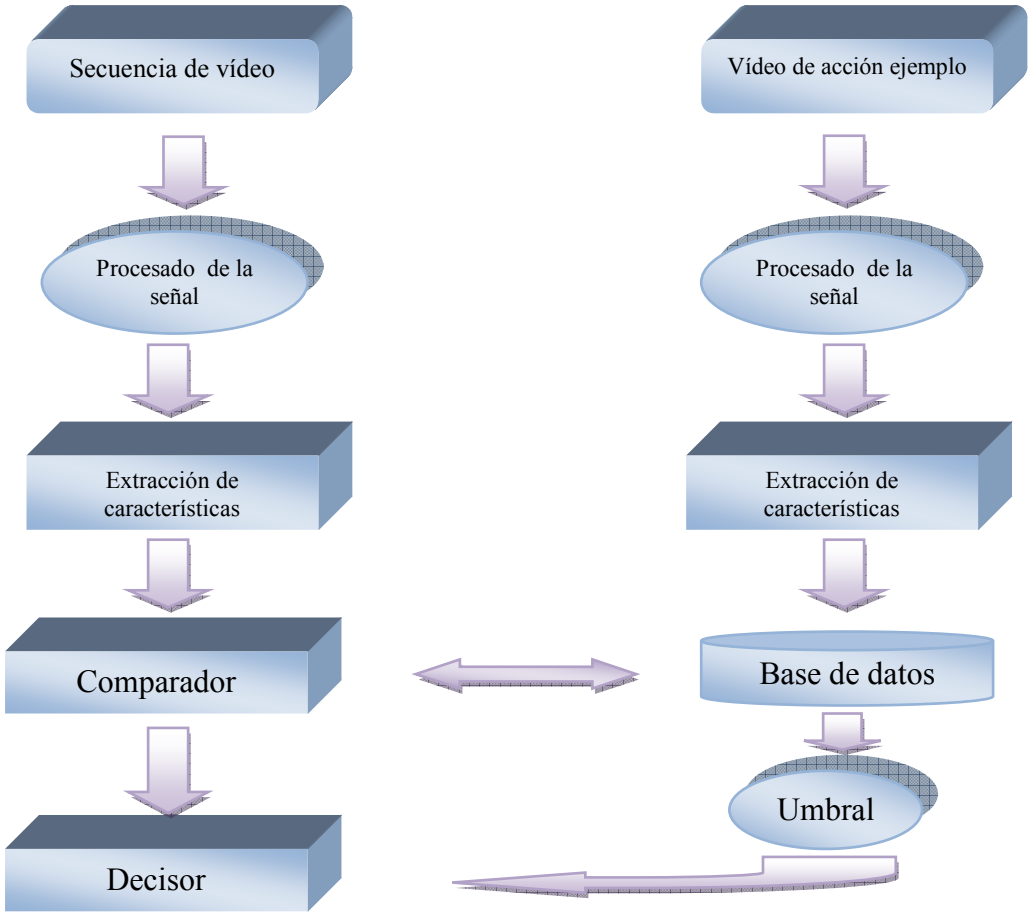


Figura 4.1: Esquema general del algoritmo.

4.3.1. Captura de la secuencia de vídeo

En esta primera fase se introducen en el sistema tanto los videos que contienen los movimientos ejemplo sobre los que queremos trabajar como las secuencias de vídeo que pretendemos analizar, en las que aparecen distintos movimientos humanos que aspiramos poder discernir. Una vez tenemos los vídeos dentro del sistema procedemos a realizar las diferentes operaciones que nos permiten trabajar en las siguientes fases, estas operaciones componen la etapa del procesado de la señal. El diseño de esta fase se puede observar en la figura 4.2.

Si estamos en el modo de registro de un movimiento ejemplo, que más adelante queremos detectar, una vez que se han recogido los datos del vídeo frame a frame, como particularidad de este modo en esta fase, se guarda como tamaño de la ventana, para luego analizar las siguientes secuencias de vídeo, el número de frames del vídeo del movimiento ejemplo.

Una vez que se han introducido los datos de la secuencia de vídeo que se pretende analizar en el sistema pasamos a la etapa de procesado de la señal, esta etapa se puede dividir en dos operaciones fundamentales, i) es la segmentación del foreground y del background y ii) realizar una pirámide temporal de la secuencia de vídeo introducida. Por un lado mediante la segmentación obtenemos una media del fondo que luego utilizaremos para separar correctamente el foreground del backgruond. Y por otro lado en cada iteración escogemos una ventana de frames de la secuencia de vídeo, a la cual aplicamos una serie de operaciones para obtener una pirámide temporal de tres niveles, para más adelante poder obtener medidas de la secuencia en diferentes escalas temporales.

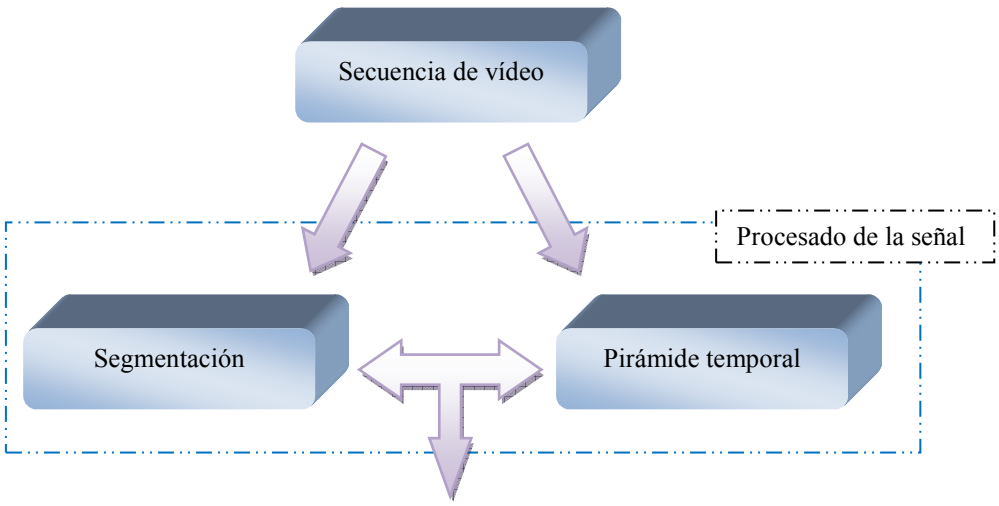


Figura 4.2: Esquema de la fase captura de la secuencia de vídeo.

4.3.2. Extracción de características

Esta segunda fase engloba la mayor parte de las operaciones que se realizan en el algoritmo, la tarea principal que se intenta desarrollar en esta etapa consiste en extraer únicamente las características que son más discriminantes entre los distintitos tipos de acciones humanas, procurando que estas características permanezcan invariantes para distintas personas que realizan el mismo movimiento. Al mismo tiempo se elimina la información de la secuencia de vídeo que no sea interesante ni útil para el proceso de reconocimiento de movimientos, como por ejemplo reduciendo la zona en cada frame donde se buscan los pixeles que contienen las características que necesitamos para caracterizar un movimiento, el hecho de eliminar esta información que no nos es útil provoca que se reduzca la duración del proceso y el coste computacional. El esquema de esta fase lo podemos encontrar en la figura 4.3.

En esta fase se trabaja fundamentalmente sobre los tres vídeos que corresponden a cada uno de los tres niveles de la pirámide temporal que hemos obtenido en la fase anterior operando sobre la ventana correspondiente de la secuencia de vídeo inicial. El primer paso consiste en calcular los gradientes en los ejes 'x', 'y' y tiempo de los vídeos de la pirámide temporal, para a continuación obtener así nueve conjuntos de matrices que seguidamente pasamos a normalizar.

En el siguiente paso se utiliza la media del fondo, obtenida en la etapa de procesado de la señal de la primera fase del algoritmo, para realizar una segmentación de los tres vídeos de la pirámide temporal mediante la cual separamos el *foreground* del *background*. Una vez hemos conseguido saber en qué zonas de cada frame se produce el movimiento podemos independizar y hacer el seguimiento de los movimientos de diferentes personas que aparezcan en la secuencia, también conseguimos reducir el área de cada frame en la que vamos a buscar los pixeles que nos interesan para caracterizar cada acción, lo que como ya hemos comentado nos va a permitir ahorrar tiempo y coste computacional.

Una vez tenemos los gradientes en los tres ejes y su normalización junto con la segmentación del foreground llegamos a la etapa en la que introducimos los datos en los histogramas que van a caracterizar el movimiento. Para ello lo que hacemos es utilizar un umbral para detectar en el conjunto de matrices del gradiente del eje del tiempo los pixeles de cada frame en los que se produce el movimiento, una vez tenemos estos pixeles escogemos sólo los que coinciden con los pixeles del foreground que hemos extraído con la segmentación. Por último introducimos en los histogramas los valores de los diferentes conjuntos de matrices de los gradientes normalizados que corresponden con la posición de los pixeles que hemos escogido. Para conseguir

caracterizar correctamente un movimiento usamos nueve histogramas uno por cada uno de los tres gradientes de los tres vídeos de la pirámide temporal.

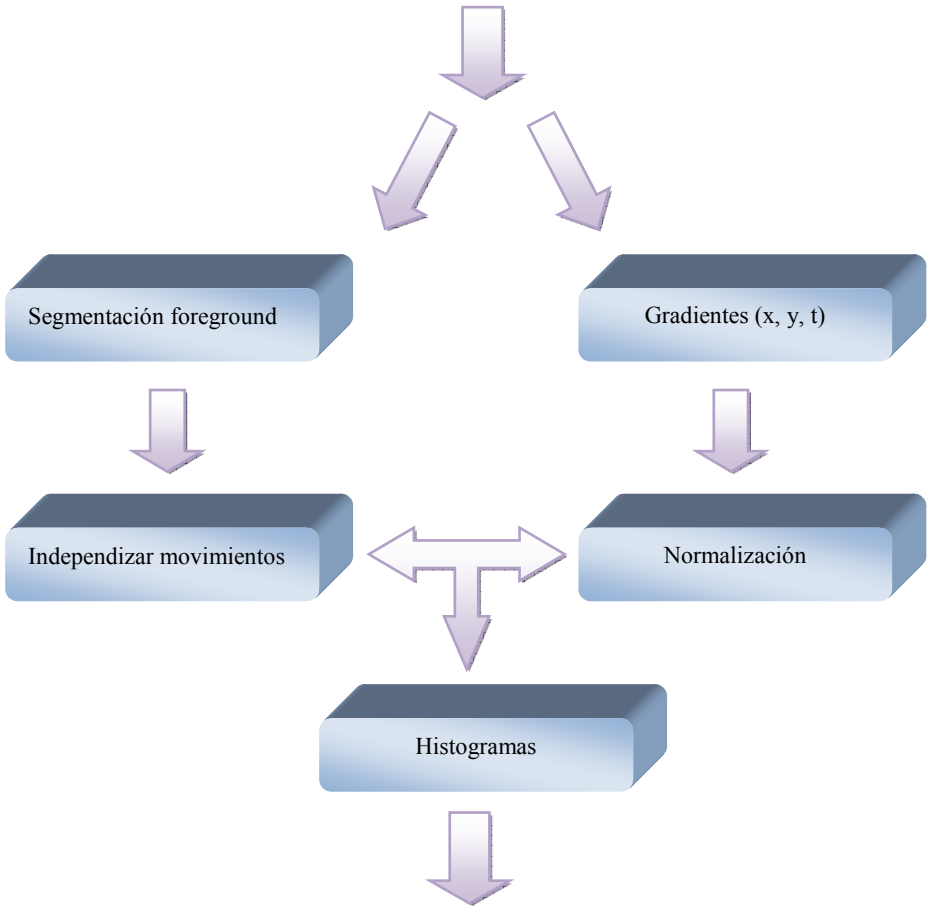


Figura 4.3: Esquema de la fase extracción de características.

4.3.3. Comparador y decisor

En esta última fase del algoritmo una vez que mediante los histogramas hemos extraído las características más significativas con las que se define un movimiento que aparece en la secuencia que estamos estudiando, estos histogramas se comparan con el conjunto de histogramas de la acción o acciones ejemplo que hemos almacenado con anterioridad en la base de datos del sistema, para poder así determinar si se trata del mismo movimiento o no. Esta comparación se realiza mediante una operación que nos da como resultado la "distancia de comportamiento", D^2 entre dos secuencias de vídeo, la fórmula utilizada es la siguiente:

$$D^2 = \frac{1}{3L} \sum_{k,l,i} \frac{[h_{1k}^l(i) - h_{2k}^l(i)]^2}{h_{1k}^l(i) + h_{2k}^l(i)}$$

K € {x, y, t}
l= 1,…, L (niveles de pirámide temporal)
i= 1,…, nº de divisiones histograma

(4.1)

Donde l es el nivel de la pirámide que estamos usando, k el eje con el que estamos trabajando e i el número de bin del histograma en el que nos encontramos; h_1 y h_2 representan los histogramas correspondientes a la acción ejemplo y al movimiento estudiado respectivamente.

Ya con esta medida de similitud entre dos secuencias lo único que falta por hacer es, utilizando un umbral, decidir si la acción que aparece el vídeo analizado corresponde con nuestra acción ejemplo y esta decisión sería nuestro resultado.

5. Desarrollo

5.1. Introducción

A lo largo de este capítulo se detalla en profundidad como se ha llevado a cabo la implementación del algoritmo descrito en esta memoria cuyo objetivo principal, consiste en el reconocimiento de actividades en videos basándose en acciones ejemplo. Como recordatorio decir que para desarrollar este algoritmo nos hemos basado inicialmente en el trabajo de Zelnik-Manor e Irani, descrito en el documento "Statistical Analysis of Dynamic Actions"[1].

Es por ello que a continuación se explicarán las decisiones que hemos tomado en cada una de las fases en base a que técnica es mejor, describiendo los criterios discriminatorios elegidos y explicando en detalle cada una de las fases del sistema de reconocimiento de acciones en video. Las tres fases principales que hemos implementado en el algoritmo son: i) en la primera fase se capturan los datos de la secuencia de video, se realiza la segmentación para así conseguir encontrar las regiones de fondo y se crea la pirámide temporal, ii) en la fase de extracción de características se generan los gradientes, se independizan los diferentes movimientos que se producen en la escena y se crean los histogramas que caracterizan las distintas acciones y iii) en la última la fase en la que se realiza la comparación y la decisión final. Como se puede observar estas tres fases son equivalentes a las que hemos comentado de forma superficial en el capítulo anterior. En la figura 5.1 podemos ver un ejemplo de lo que queremos conseguir con este sistema.

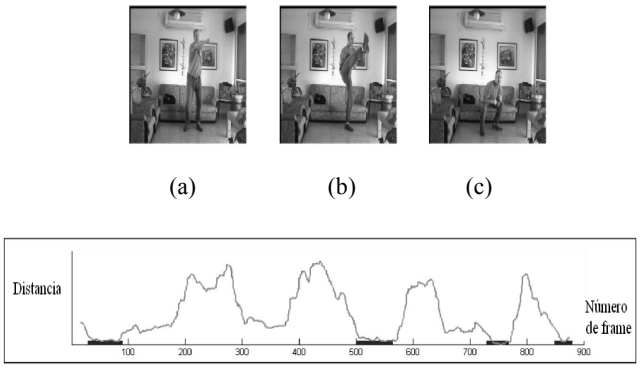
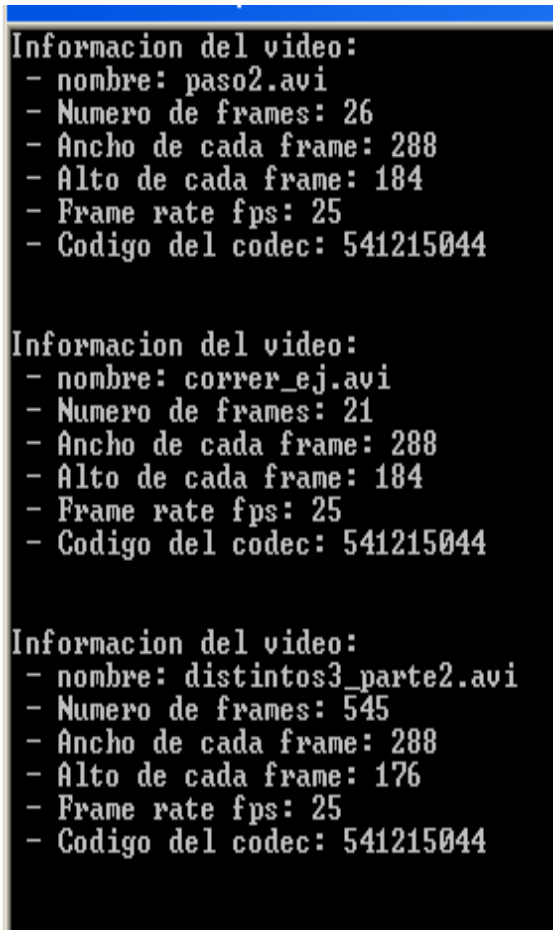


Figura 5.1: En las tres primeras imágenes se observan las acciones puñetazo, patada y agacharse, que son los tres movimientos que podemos ver en el vídeo. La gráfica muestra la medida de distancia entre un ejemplo de la acción puñetazo y el resto de subsecuencias.

5.2. Captura de la secuencia de vídeo

5.2.1. Captura de los datos de la secuencia de vídeo

Al comenzar el algoritmo lo primero que hacemos es obtener la información general que describe cada secuencia de vídeo, concretamente la información que se busca es la siguiente (Figura 5.2):



- ***Nombre de la secuencia de vídeo:*** se utilizara para saber con que vídeo estamos trabajando y para dar nombre a los ficheros y vídeos de los correspondientes resultados.

- ***Número de frames:*** Este dato se utilizara para generar los gradientes, para realizar la segmentación, para el cálculo de los histogramas, etc. En general en casi todo el sistema.

- ***Alto y ancho de cada frame:*** Cada una de las matrices que utilizaremos más adelante tendrán los mismos valores de acho y alto que los de la secuencia inicial.

- ***La tasa de frames:*** Los vídeos resultado tendrán la misma tasa de frames.

- ***Código de códec:*** Los vídeos resultado se guardaran con la misma codificación.

Figura 5.2: Ejemplo de la información que obtenemos de las secuencias de vídeo que introducimos en el sistema. En este ejemplo se muestra en los dos primeros bloques la información de dos secuencias que contienen movimientos ejemplo y en el tercer bloque la información de la secuencia que queremos analizar.

Justo después de obtener esta información del vídeo que queremos analizar, pasamos a capturar los datos de todos los frames de la secuencia, guardándolos en diferentes conjuntos de matrices para su posterior utilización durante el algoritmo. Para esta tarea necesitamos cuatro conjuntos de matrices.

Para rellenar el primer conjunto de matrices guardamos los datos de la secuencia una vez que es convertida a escala de grises, este conjunto lo usaremos más adelante para la segmentación del fondo, para independizar los movimientos y para generar el vídeo resultado. En los tres conjuntos de matrices restantes guardamos los datos de cada una de las tres componentes primarias de color (RGB) de la secuencia de vídeo, es decir en un conjunto se guardan los datos que corresponden a los rojos, en otro los datos que corresponden a los verdes y en el tercero los que corresponden a los azules, estos tres conjuntos de matrices se usaran posteriormente para generar las diferentes escalas temporales de la secuencia con las que vamos a trabajar y para calcular los gradientes.

5.2.2 Segmentación para la substracción del fondo

Un punto importante en el procesado de la señal es el de la segmentación para la substracción del fondo, ya que nos va a ser muy útil en los pasos posteriores del algoritmo. En esta etapa inicializamos y representamos un modelo de fondo robusto de la secuencia de video analizada, para conseguir esto el fondo se describe mediante un modelo matemático para cada píxel de la imagen en cada instante de tiempo.

El modelo o método que hemos escogido para representar el fondo consiste en el método de la media o mediana, calculamos el fondo como la media de 'n' frames (normalmente escogemos entre 100-150 frames) de la secuencia, cada 'n' frames volvemos a calcular una nueva media del fondo. Si el número de frames totales de la secuencia no es superior a '2n' frames o es inferior a 'n' frames se calcula la media del fondo con la totalidad de los frames del vídeo. Si además después de hacer la última media del fondo el número de frames que restan de la secuencia es inferior a 'n' frames, estos frames restantes también se usaran para hacer la última media del fondo. En la figura 5.3 se muestra algunos de los resultados que obtenemos al usar este método de substracción del fondo.

$$Bg(x,y) = \frac{1}{N}\sum_{i=1}^{N} I_i(x,y) \qquad (5.1)$$

La utilización de este método en concreto se debe a que normalmente en la mayoría de los vídeos que vamos a analizar con este sistema el tipo de fondo que aparece es unimodal (Figura 5.3 a) y b)), en este tipo de fondo los valores de los pixeles del fondo no cambian a lo largo de toda la secuencia, la cámara suele ser estática y no se producen variaciones en la iluminación. Aunque también el sistema funciona con algunos fondos multimodales (Figura 5.3 c)) en los que la cámara permanece estática y no hay cambios en la iluminación, pero si existen objetos que pertenecen al fondo que están en leve movimiento (como olas de mar) o hay cambios en la geometría del fondo (aparición de

nuevos objetos estáticos), este último problema se soluciona gracias a que vamos actualizando el fondo cada 'n' frames.

(a) (b) c)

Figura 5.3: Tres ejemplos de substracción de fondo en tres secuencias de vídeo diferentes en las que trabajamos con distintos entornos.

En esta misma etapa además de la media del fondo también aprovechamos para calcular cada 'n' frames la desviación típica del fondo, la cual almacenaremos para usarla en una etapa futura en la que debemos hacer la detección de frente, ya que para detectar las zonas en las que se produce el movimiento utilizaremos técnicas basadas en gaussiana y para ello necesitaremos tanto la media del fondo como su desviación típica.

5.2.3. Creación de la pirámide temporal

Por otro lado para poder comparar nuestra secuencia con los diferentes movimientos ejemplo debemos ir analizando el vídeo escogiendo una ventana de 'M' frames en cada iteración. Estos 'M' frames corresponden al número de frames que componen la secuencia de vídeo que contiene el movimiento ejemplo que estemos estudiando. En cada nueva iteración para conseguir la siguiente ventana desechamos el primer frame del conjunto anterior y añadimos un nuevo frame al final. Vamos a trabajar con cada una de las subsecuencias que obtenemos y sobre cada una de ellas vamos a crear una pirámide temporal.

El hecho de crear la pirámide temporal de cada subsecuencia se debe a que nos permite comparar una misma acción en diferentes escalas temporales. Normalmente una acción específica se ejecuta con aproximadamente la misma velocidad, por lo tanto para vídeos con la misma tasa de frames y sin tener en cuenta el resto de parámetros, una misma acción que realizan dos personas distintas o es realizada por una persona pero captada por dos cámaras distintas se extenderá de forma aproximada por el mismo número de frames. Esto nos indica que normalmente podremos diferenciar dos actividades sin tener que utilizar diferentes escalas temporales, aunque el hecho de utilizar diferentes escalas temporales nos favorece a la hora de obtener una mejor medida de similitud entre dos acciones. Sin embargo, la comparación a través de escalas temporales es imprescindible

si queremos poder diferenciar acciones cuando estas son capturadas en múltiples escalas temporales, es decir, si una misma actividad se realiza con velocidades significativamente diferentes. También es posible que aún ayudándonos del uso de diferentes escalas temporales no consigamos distinguir correctamente dos acciones distintas debido a su similitud tanto en velocidad como en movimiento, como por ejemplo distinguir entre marchar (andar rápidamente) y hacer footing.

Para obtener las múltiples escalas temporales que constituyen la pirámide temporal el proceso que utilizamos consiste en hacer un muestreo de la subsecuencia del vídeo inicial y seguidamente hacer un filtrado paso-bajo a lo largo de la dirección temporal. La pirámide temporal (Figura 5.4) de una secuencia S es una pirámide de secuencias S^1(=S); S^2;...; S^L, donde los frames de imagen en todos los niveles (secuencias) dentro de la pirámide son del mismo tamaño, y cada secuencia S^l tiene la mitad del número de frames de la secuencia de resolución inmediatamente anterior S^{l-1}.

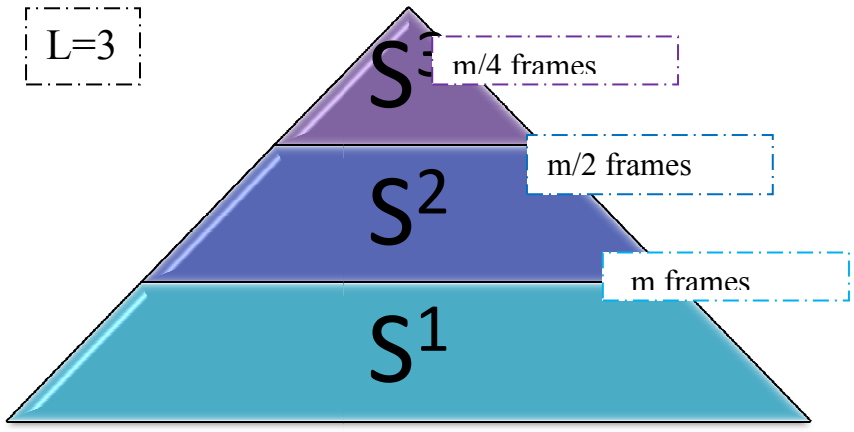


Figura 5.4: Esquema de la pirámide temporal que vamos a utilizar durante el algoritmo.

5.3. Extracción de características

5.3.1. Cálculo de los gradientes

En este apartado vamos a explicar el proceso por el cual calculamos los gradientes en los ejes 'x', 'y' y 't' de los conjuntos de matrices que representan la secuencia. El gradiente denota una dirección en la que se aprecia una variación de una determinada propiedad o magnitud física, en nuestro caso el gradiente en los ejes 'x' e 'y' nos servirá para identificar los cambios de los valores de los píxeles de un mismo frame en el espacio, y el gradiente en el tiempo nos aportará información de cómo varían los píxeles

a lo largo de diferentes frame, con lo que podremos detectar el movimiento que se produce en la secuencia. Esta etapa es fundamental dentro de nuestro sistema ya que de ella obtendremos la mayor parte de los datos que nos van a ayudar a caracterizar las diferentes acciones y movimientos.

En este punto vamos a trabajar con los tres conjuntos de matrices que representan las tres componentes primarias de color (rojo, verde, azul) de la subsecuencia de tamaño 'M' frames que estamos analizando, a las cuales hemos aplicado un proceso para generar una pirámide temporal de cada uno de los tres conjuntos. Por lo tanto usaremos nueve vídeos, tres por cada nivel de la pirámide temporal.

Para cada nivel (la secuencia) S^l de la pirámide temporal, estimamos los gradientes de intensidad espacio-temporal local (S^l_x; S^l_y; S^l_t) en todos los puntos espacio-temporales. Para conseguir esto aplicamos a cada uno de los tres conjuntos de matrices, que representan las componentes primarias de color de un nivel de la pirámide temporal, tres filtros de Sobel de tres dimensiones de forma independiente, uno por cada eje (eje 'x', eje 'y' y eje tiempo), los filtros son los siguientes (Figura 5.5):

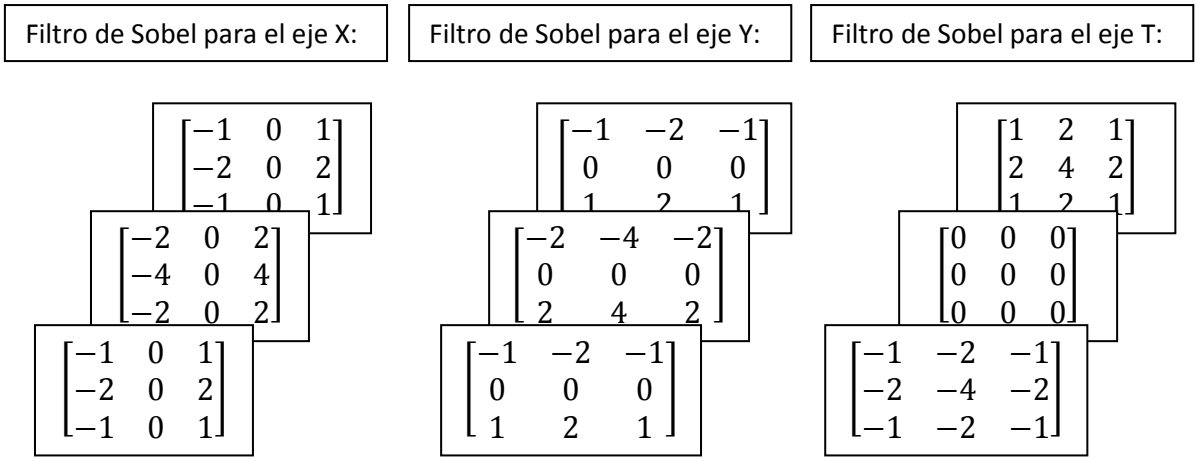


Figura 5.5: Filtros de Sobel.

Inmediatamente después de aplicar los filtros de Sobel obtendremos como resultado tres conjuntos de matrices uno por cada componente primaria de color por cada uno de los ejes de gradiente, entonces para rellenar cada valor del conjunto de matrices que representan cada gradiente de un nivel de la pirámide temporal (Figura 5.6) hacemos la siguiente selección: escogemos para cada uno de los valores de gradiente, que representaría cada pixel de los frames de un gradiente, el valor correspondiente que sea mayor entre los tres conjuntos de matrices de componente de color primaria que hemos obtenido por cada gradiente. Esto nos da como ventaja tener un mayor contraste entre el primer plano y el fondo logrando una mayor información conductual.

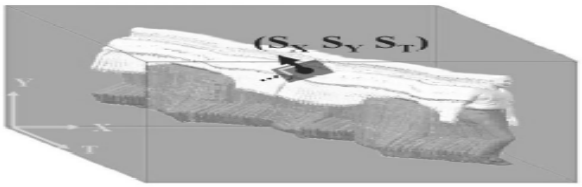


Figura 5.6: Volumen espacio-temporal S correspondiente a una persona caminando. El gradiente espacio-temporal (S_x , S_y , S_t)se estima en cada punto espacio-temporal (x, y, t).

El gradiente es normal a la superficie espacio-temporal local generada según el movimiento en el volumen espacio-temporal de la secuencia (en la resolución temporal l). Así, la dirección de gradiente captura la orientación superficial local, que depende sobre todo de las propiedades conductuales locales del objeto móvil, mientras su magnitud depende principalmente de las propiedades fotométricas locales del objeto móvil y es afectada por su aspecto espacial (p.ej., el contraste, el color, la textura de la ropa, la iluminación, etc.).

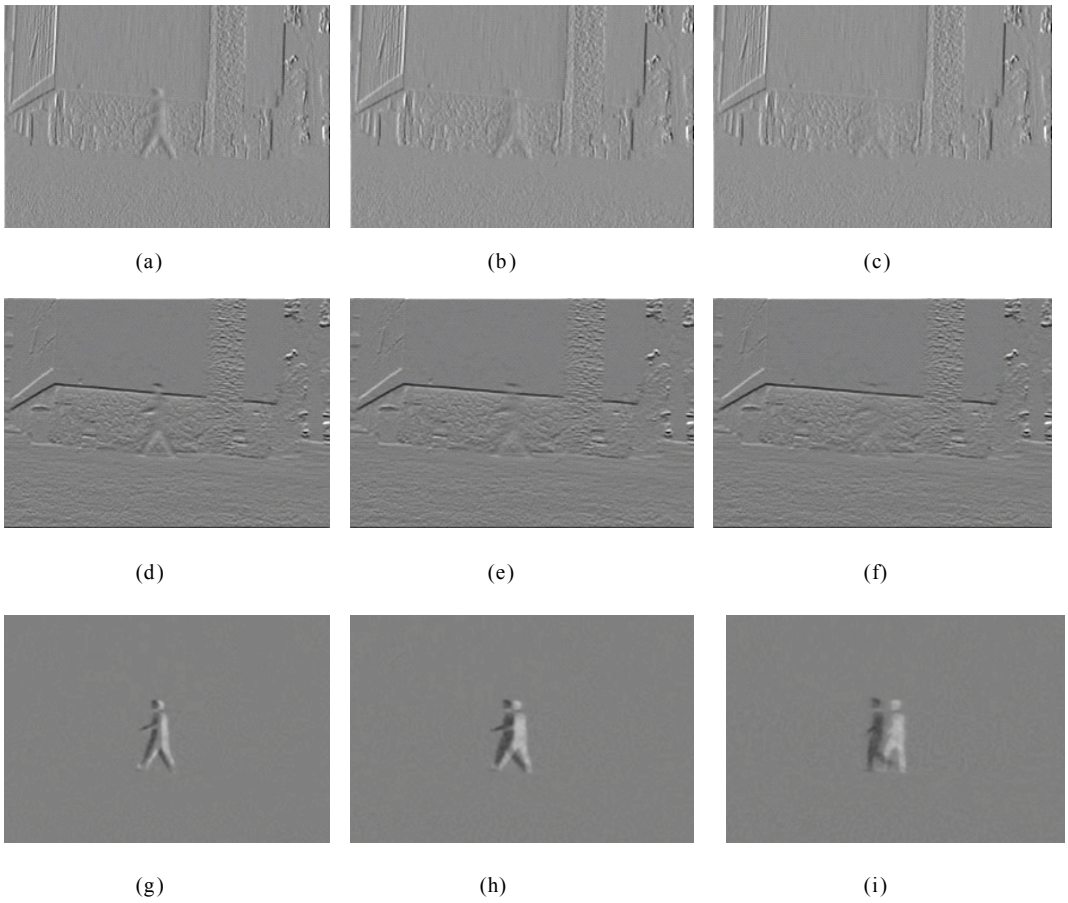


Figura 5.7: En (a), (b) y (c) muestran ejemplos de un frame del gradiente en el eje 'x', cada imagen corresponde a un nivel de la pirámide temporal para la acción de caminar. En (d), (e) y (f) se muestra el mismo ejemplo pero del gradiente en el eje 'y'. Por último (g), (h) y (i) muestran el mismo ejemplo pero del gradiente en el eje del tiempo.

Una vez hemos obtenido los nueve conjuntos de matrices que corresponden a los gradientes de intensidad espacio-temporal (figura 5.7) con los que podemos representar un movimiento, pasamos a normalizarlos para intentar conseguir conservar solamente la información conductual (orientación) y eliminar tanta componente fotométrica como sea posible (la magnitud). Para que el modelo sea invariante a la dirección de la acción (por ejemplo de derecha a izquierda o de izquierda a derecha), tomamos el valor absoluto de los gradientes normalizados. En la fórmula 5.2 se describe la normalización de la que haremos uso.

$$\left(N_x^l,\ N_{y,}^l\ N_t^l\right) = \frac{(|S_x^l|,|S_y^l|,|S_t^l|)}{\sqrt{(S_x^l)^2 + (S_y^l)^2 + (S_t^l)^2}} \qquad \begin{matrix} l=1,\ldots, L \\ L=3 \end{matrix} \qquad (5.2)$$

Donde l es el nivel de la pirámide temporal, S^l_x; S^l_y; S^l_t son los gradientes de intensidad espacio-temporal en cada uno de los posibles niveles de la pirámide temporal y N^l_x; N^l_y; N^l_t corresponden a los gradientes espacio-temporales normalizados.

Nuestra representación de acción consiste en una distribución 3L-dimensional del conjunto de medidas de cada punto espacio-temporal a través de todas las escalas temporales. Por lo tanto cada punto espacio-temporal es asociado con vector 9-dimensional: [N1x; N1y; N1t; N2x; N2y; N2t; N3x; N3y; N3t].

5.3.2. Segmentación para la detección del frente

En este punto se explicará de forma detallada los pasos del proceso por el cual conseguimos detectar las zonas de la imagen en donde se produce el movimiento, pudiendo así independizar las actividades que realiza cada persona que aparece en la secuencia de vídeo, para que más adelante podamos estudiar su similitud con las acciones ejemplo.

Para la consecución de esta tarea necesitamos recuperar la media y la desviación típica que hemos obtenido en la primera etapa del algoritmo y trabajaremos sobre el conjunto de matrices en el que hemos guardado los datos de la secuencia de vídeo en escala de grises.

La técnica que utilizamos para lograr localizar las zonas pertenecientes al frente es el método basado en Gaussiana, por el cual comparamos cada matriz, que representa un frame de la subsecuencia que estamos analizando, con el modelo de fondo de distribución Gaussiana, que obtenemos gracias a la media y a la desviación típica del fondo (μ, σ2), midiendo así la probabilidad en luminancia. Una vez hecha esta

comparación podemos decidir que pixeles corresponden al fondo, a los cuales le damos el valor 255 (negro), y que pixeles pertenecen al frente a los que les daremos el valor 0 (blanco) (Figura 5.8). Para que podamos clasificar un pixel como integrante del frente debe cumplir la siguiente condición descrita a continuación en la fórmula 5.3:

$$\mu(x,y) - \sigma(x,y) < I_t(x,y) < \mu(x,y) + \sigma(x,y) \Rightarrow F_t(x,y) = 0\ (blanco) \qquad (5.3)$$

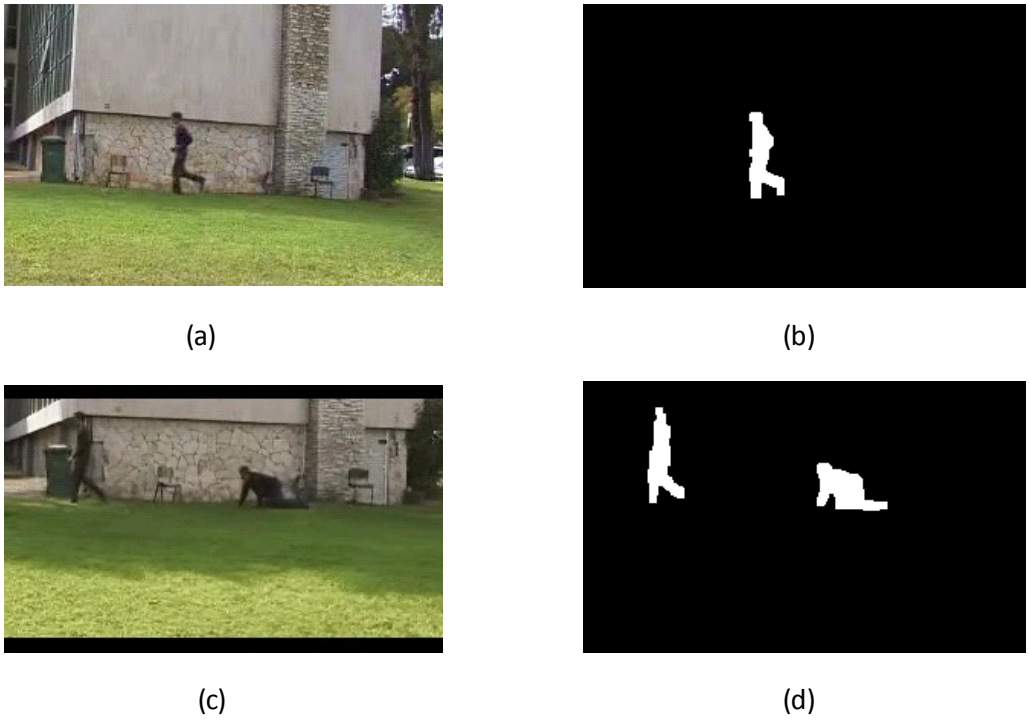


Figura 5.8: En (a) y (b) se muestra un ejemplo de una persona corriendo y su respectivo resultado de la detección de frente. En (c) y (d) se muestra un ejemplo en el que aparecen dos personas simultáneamente en la secuencia una caminando y otra gateando y el correspondiente resultado de la detección de frente.

El siguiente paso que debemos realizar en este proceso una vez hemos detectado las zonas en las que se produce el movimiento es el seguimiento de las diferentes personas que se encuentran en la secuencia, de esta forma podremos observar de manera independiente las distintas actividades del vídeo en cuestión.

Lo primero que hemos hecho para efectuar esta tarea es crear una función que nos permite hallar los contornos de las figuras que forman los pixeles blancos unidos y luego generamos por cada figura un rectángulo que la contiene Una vez generados estos rectángulos escogemos solamente los que son suficientemente grandes como para contener una silueta de una persona (en este caso se ha decidido que sean rectángulos

mayores de 30 pixeles de ancho y 40 pixeles de alto), estos rectángulos que se eligen como útiles se guardan en un array de rectángulos. En la figura 5.9 se observa un ejemplo de cómo independizamos los movimientos dentro de la misma escena.

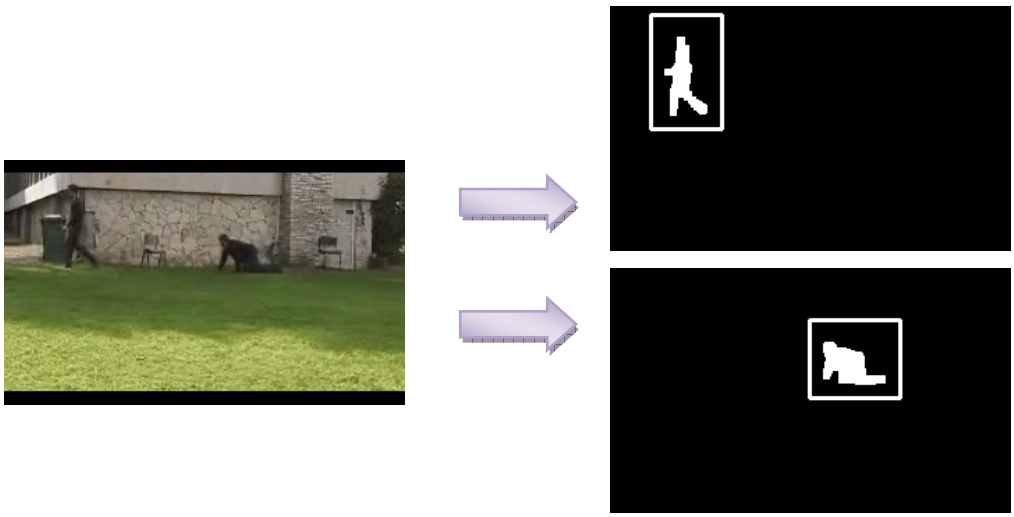


Figura 5.9: Se muestra un ejemplo de cómo se consigue independizar cada acción de la escena acotando las zonas donde se produce el movimiento mediante estos rectángulos.

Con el conjunto de rectángulos que aparecen por cada frame vamos rellenando un array de una estructura que se ha creado específicamente para poder guardar y clasificar estos rectángulos, para así poder identificar con que grupo de rectángulos se corresponde cada movimiento dentro del video. La estructura en la que guardamos dichos rectángulos obtenidos se compone de los siguientes campos:

```
typedef struct {
  int num_frames; -> número total de frames de la subsecuencia en los
  que se da determinado movimiento.
  int  *NF;  ->  guardo  el  número  de  frame  que  corresponde  a  cada
  rectángulo del grupo.
  int *x; -> coordenada inicial x de cada rectángulo del grupo.
  int *y; -> coordenada inicial y de cada rectángulo del grupo.
  int *ancho; -> ancho de cada rectángulo del grupo.
  int *alto; -> alto de cada rectángulo del grupo.
  int flag_libre; -> indica si ha sido guardado algún rectángulo en
este grupo (0= no hay ningún rectángulo, grupo libre; 1= grupo en el
que ya ha sido guardado algún rectángulo).
  int distancia; -> el valor de la distancia que obtengo más adelante
con respecto al movimiento que representa este grupo de rectángulos.
} GRUPO_RECT;
```

En principio hemos decidido que este array tenga capacidad para guardar diez estructuras de este tipo, es decir, diez movimientos distintos como máximo, esto se ha hecho de esta forma porque por ahora en los videos con los que trabajamos el número de actividades distintas que aparecen a la vez es bastante bajo, normalmente entre una y

cinco. Como aclaración decir que en cada grupo de rectángulos se guarda como mucho un rectángulo por cada frame de la secuencia.

Para rellenar correctamente el array de grupos de rectángulos lo primero que hacemos con cada uno de los rectángulos de cada frame es ver si pertenece a alguno de los grupos de rectángulos que ya estén iniciados, es decir, los grupos en los que ya se haya guardado algún rectángulo. Un rectángulo pertenece a un grupo ya iniciado si cumple la condición de que su centro este próximo al centro del último rectángulo que se haya guardado en el grupo. Si el nuevo rectángulo no cumple esta condición con ninguno de los grupos iniciados quiere decir que corresponde a un grupo nuevo y se introduce en un grupo con el flag libre a cero, que indica que es un grupo vacio.

Después de ubicar cada uno de los rectángulos en su respectivo grupo, descartamos aquellos grupos que tienen rectángulos en menos de tres frames del video, porque suelen indicar zonas en las que aparecen siluetas debido a una mala segmentación, o porque hay un cambio en el fondo del video, y como aún no se ha actualizado de nuevo la media del fondo, aparecen rectángulos no deseados, o porque un movimiento en menos de tres frames es tan rápido y tiene tan poca información que se descarta.

Gracias a este proceso logramos independizar las actividades de las diferentes personas que aparecen en la secuencia de vídeo con la que estamos trabajando, lo que nos permite en los pasos posteriores conseguir para cada acción un conjunto de histogramas que la caractericen, con los que podremos realizar la comparación con las acciones modelo de nuestra base de datos.

5.3.3. Cálculo de los histogramas

Ahora pasaremos a describir otra de las fases fundamentales de este algoritmo, en este caso vamos a hablar de cuál es el método que aplicamos para introducir los datos que nos interesan para caracterizar un movimiento en un conjunto de histogramas, que usaremos más adelante para examinar el grado de similitud entre dos actividades. De esta etapa debemos obtener como resultado un grupo de nueve histogramas uno por cada eje de coordenadas en cada uno de los tres niveles de la pirámide temporal que tenemos de cada subsecuencia del video que hemos introducido en el sistema.

En esta fase trabajaremos sobre los resultados que hemos obtenido en las fases anteriores, tanto sobre los nueve conjuntos de matrices que representan a los gradientes y sus respectivas normalizaciones, como sobre los resultados que hemos obtenido al independizar las distintas acciones que aparecen en la secuencia.

Inicialmente para representar un movimiento independiente introducimos un único grupo de rectángulos en la función que calcula los histogramas. A continuación buscamos en cada frame del gradiente del eje de tiempo en las zonas que nos indican los rectángulos la posición de los pixeles en los que se produce el movimiento. Para escoger los pixeles que nos interesan utilizamos dos umbrales, de esta forma seleccionamos como pixeles válidos aquellos que sean mayores que el primer umbral o los que sean menores que el segundo umbral. El hecho de buscar las posiciones de los pixeles útiles en las zonas acotadas por los rectángulos obtenidos en la fase anterior del algoritmo, nos permite reducir el ruido que podemos introducir en los histogramas y reducir también tanto el coste computacional como el tiempo de ejecución. Por último, para que un pixel sea válido además de pertenecer al conjunto de pixeles que indican movimiento dentro de los frames del gradiente en el eje temporal, debe coincidir con un pixel blanco que indique que pertenece al frente en el frame correspondiente de la segmentación que realizada en el apartado anterior. En la figura 5.10 se muestra un diagrama en el que observamos la manera en la que obtenemos los píxeles que nos interesan.

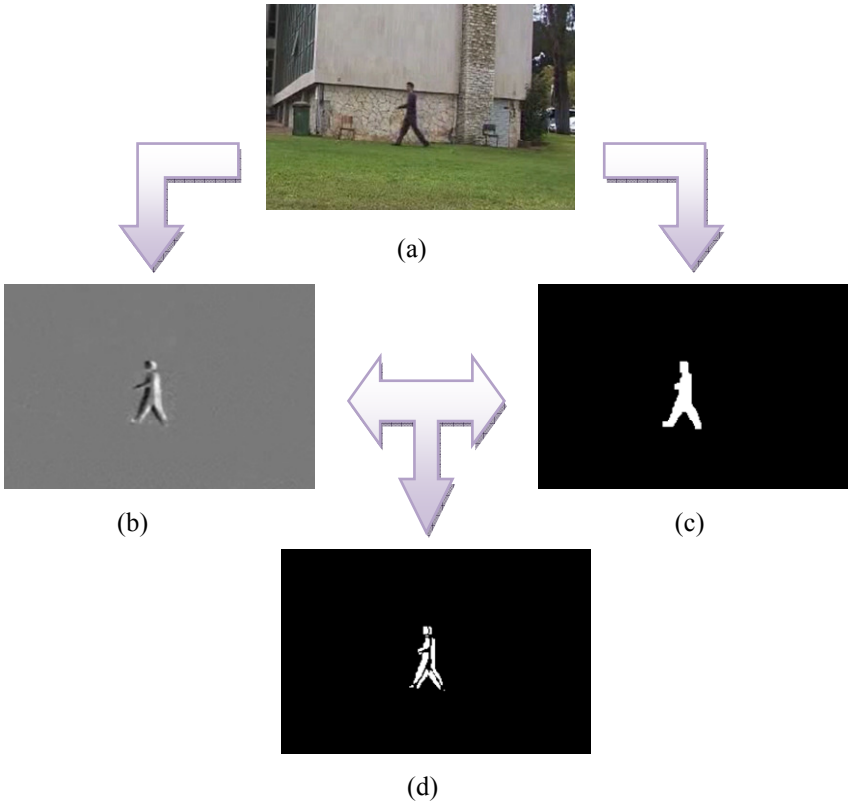


Figura 5.10: Diagrama de flujo que nos enseña cómo obtenemos la posición de los pixeles que nos interesan para una secuencia en la que aparece una persona caminando.(a)Frame sobre el que estamos trabajando. (b) Resultado de calcular el gradiente temporal sobre el frame inicial. (c)Resultado de la segmentación del frame inicial. (d) Obtención de los pixiles útiles.

Tras finalizar la tarea de extarción e indentificación de las posiciones de los pixeles de cada frame de la secuencia que nos interesan para caracterizar un movimiento, buscamos a partir de las mismas el valor correspondiente en el conjunto de matrices que representan a los gradientes normalizados (Nx, Ny, Nt), y finalmente estos valores los introducimos en su respectivo histograma, uno de los tres posibles que existen por cada escala temporal de la subsecuencia. De esta forma logramos obtener los nueve histogramas con los que podemos caracterizar correctamente una acción humana concreta (h1x; h1y; h1t; h2 x; h2y; h2t; h3x; h3y; h3t).

Los datos de los histogramas que caracterizan cada movimiento son guardados en la siguiente estructura:

```
typedef struct {
  int hist_bins_x1[256];
  int hist_bins_y1[256];
  int hist_bins_t1[256];
  int hist_bins_x2[256];
  int hist_bins_y2[256];
  int hist_bins_t2[256];
  int hist_bins_x3[256];
  int hist_bins_y3[256];
  int hist_bins_t3[256];
} HISTOGRAMAS;
```

El hecho de que representemos una acción usando estos nueve histogramas se debe a que en una etapa previa del proyecto se descartó representar las distintas acciones mediante histogramas multidimensionales, ya que son computacionalmente costosos y consumen mucha memoria. Así que decidimos asumir que todos los componentes de un punto espacio-temporal son independientes el uno del otro. Representamos cada acción con un conjunto de 3L distribuciones unidimensionales, una para cada componente de las medidas espacio-temporales (x,y,t) en cada uno de los niveles de la pirámide temporal. En la figura 5.11 se muestra un ejemplo de los histogramas que logramos en el primer nivel de la pirámide temporal para la acción del tipo correr.

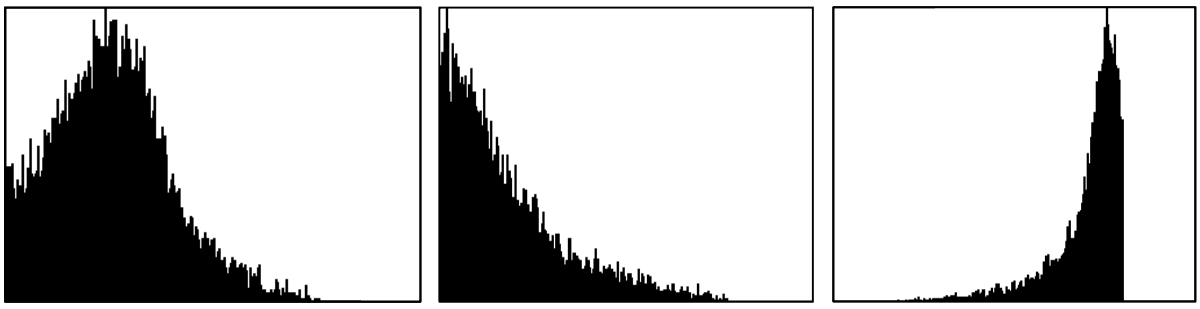


Figura 5.11: Histogramas en cada una de las componentes 'x', 'y' y 't' del primer nivel de la pirámide temporal para un ejemplo de la acción del tipo correr.

5.4. Comparación y decisión

5.4.1. Comparación

Esta sección se describe la fase definitiva de nuestro algoritmo en la que apoyándonos sobre el conjunto de histogramas que representan cada actividad humana, vamos a conseguir estudiar su similitud conductual con respecto a las acciones que hemos escogido como ejemplo o modelo. En la figura 5.12 podemos comprobar las diferencias que existen entre los histogramas pertenecientes a distintos movimientos

Una vez que hemos recogido las características que definen un movimiento, que está presente en la secuencia de vídeo que estamos analizando, dentro de su respectivo conjunto de nueve histogramas, pasamos a realizar una comparativa con la actividad o actividades humanas ejemplo que hemos guardado con anterioridad en nuestra base de datos mediante sus respectivos conjuntos de histogramas que las caracterizan. Para realizar esta comparativa entre dos acciones vamos a utilizar la siguiente fórmula 5.4, que nos da como resultado la distancia de "comportamiento" entre dos secuencias.

$$D^2 = \frac{1}{3L} \sum_{k,l,i} \frac{[h^l_{1k}(i) - h^l_{2k}(i)]^2}{h^l_{1k}(i) + h^l_{2k}(i)} \qquad (5.4)$$

K € {x, y, t}
l= 1,…, L (niveles de pirámide temporal)
i= 1,…, nº de divisiones histograma

Donde l es el nivel de la pirámide que estamos usando, k el eje con el que estamos trabajando e i el número de bin del histograma en el que nos encontramos; h_1 y h_2 representan los histogramas correspondientes a la acción ejemplo y al movimiento estudiado respectivamente.

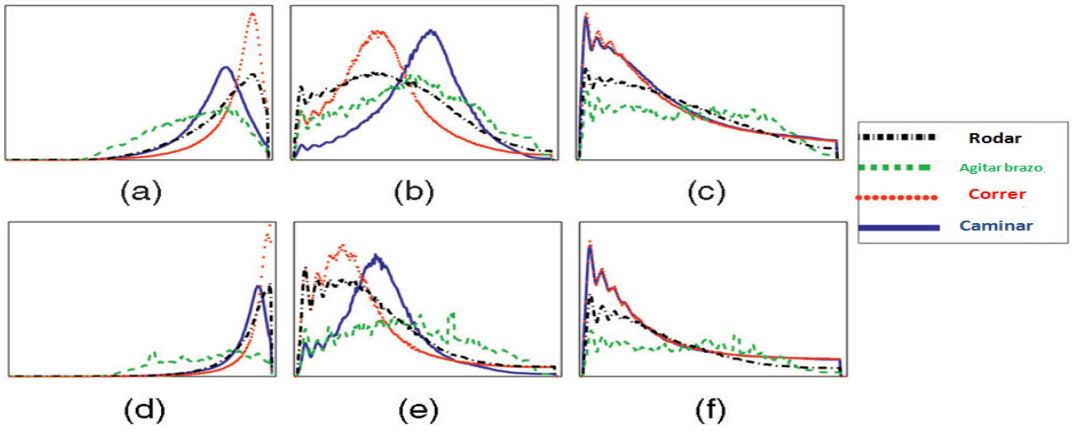


Figura 5.12: Comparativa de los histogramas de 4 acciones distintas, que son correr, caminar, agitar los brazos y rodar. (a)h^1_x, (b)h^1_y, (c)h^1_t, (d)h^2_x, (e)h^2_y y (f)h^2_t.

5.4.2. Decisión

Una vez calculada esta medida de distancia entre las acciones que aparecen en dos secuencias de vídeo distintas, el último paso que debemos dar en este algoritmo consiste en tomar la decisión de si se trata de la misma acción o no, para conseguir esto empleamos un umbral. Si la distancia entre nuestra actividad y la actividad ejemplo es menor o igual al umbral que estamos usando tomaremos la decisión de que nuestra actividad puede considerarse similar a la acción ejemplo, en caso contrario tomaremos la decisión opuesta.

Como nuestra representación está basada en distribuciones, se comporta de la misma forma para secuencias que muestran la misma acción, incluso cuando los tamaños de los frames son distintos o cuando se muestra un número diferente de repeticiones por acción.

En la siguiente sección demostraremos que un movimiento se puede caracterizar por un conjunto de distribuciones unidimensionales, y que la diferencia entre dos de estos conjuntos es suficiente como para discernir dos tipos de acción humana.

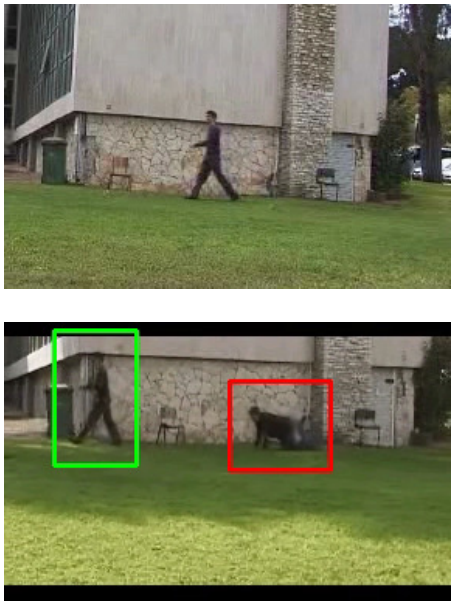


Figura 5.13: Ejemplo de un frame de una secuencia de vídeo analizada por el sistema para el caso en el que la acción modelo es caminar.

6. Experimentos y resultados

6.1. Modo de evaluación del rendimiento

Para evaluar de forma correcta los resultados obtenidos de nuestro sistema reconocedor de actividades humanas en vídeo, necesitamos una serie de herramientas y procedimientos que nos permitan evaluar su rendimiento. Para ello utilizaremos una serie de valores y curvas que nos permitan analizar diversos aspectos del algoritmo.

Desde el primer momento debemos tener en consideración que dos muestras de un mismo movimiento realizado por distintas personas (cambios en el comportamiento) o en distintas condiciones ambientales (iluminación, condiciones de cámara, etc.) no son exactamente iguales y presentan una serie de diferencias. Por ello la respuesta del comparador de nuestro sistema se basa en cuantificar la distancia conductual entre la secuencia introducida y el patrón de la base de datos con el que se está comparando. Cuanto mayor sea la similitud entre muestras, menor será la puntuación de distancia devuelta por el comparador y más seguro estará el sistema de que dos movimientos son iguales.

La decisión del algoritmo viene dada mediante un umbral, las actividades que nos den como resultado de la comparación una distancia menor que el umbral se consideraran que son iguales a las del correspondiente ejemplo que estemos usando, mientras si la distancia es superior al umbral, consideraremos estas actividades como distintas a la actividad ejemplo correspondiente.

6.1.1. Métodos de evaluación de sistemas

Si estuviésemos trabajando con un sistema ideal, los rangos de variación de las medidas de distancia obtenidas para las acciones deberían estar suficientemente separadas, de forma que no haya ningún tipo de solapamiento entre las distribuciones de las distintas actividades. De esta forma podemos establecer un umbral de decisión que discrimine perfectamente cada acción. Sin embargo, puede darse el caso en que exista una región en la que se solapen las distribuciones, interpretando toda medida de distancia bajo el umbral como movimientos iguales a los de la base de datos. Esto nos lleva a deducir

que el área bajo la curva de acciones incorrectas que queda por debajo del umbral es la probabilidad de que tomemos una acción incorrecta como válida, lo que denominaremos tasa de falsa aceptación (FAR, False Acceptance Rate). De la misma manera el área bajo la curva de movimientos correctos que queda por encima del umbral es la probabilidad de que tomemos una acción correcta como no válida, es deir, tasa de falso rechazo (FRR, False Rejection Rate). Según coloquemos el umbral la FAR y la FRR varían en sentido opuesto (Figura 6.1), si el umbral es alto nuestro sistema será muy permisivo, en caso contrario si el umbral es bajo el sistema pasará a ser muy restrictivo.

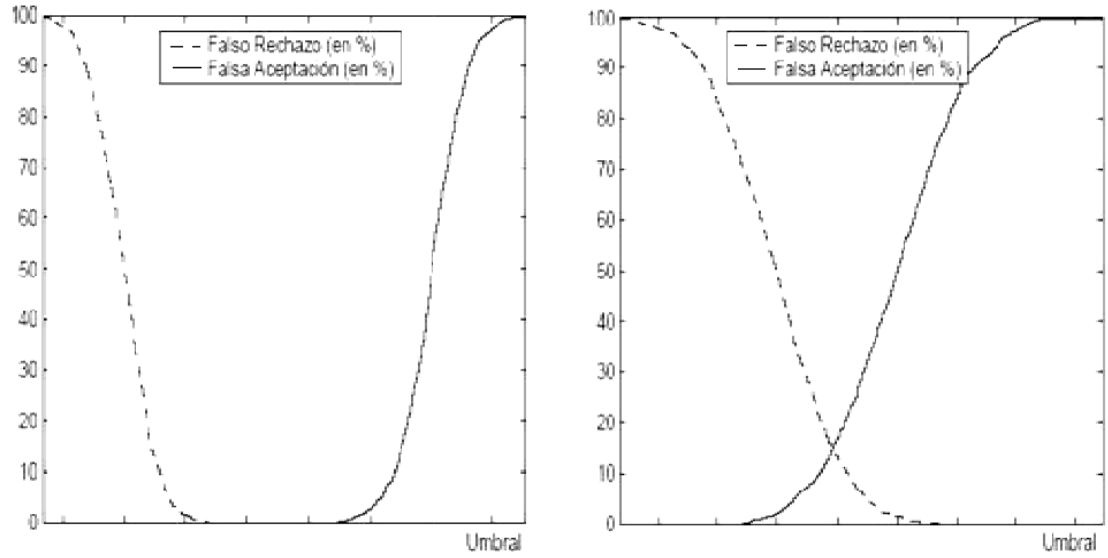


Figura 6.1: Ejemplos de FAR y FRR.

Un punto de dichas graficas que nos permite caracterizar de forma directa el funcionamiento del sistema es el punto de igual error, EER (Equal Error Rate) (Figura 6.2). Punto en que las curvas de falsa aceptación (FA) y falso rechazo (FR) se cruzan, es por ello la tasa de igual error (EER) suele usarse para caracterizar con un único número el rendimiento de un sistema.

A pesar de que el punto de EER corresponde al umbral donde se igualan FA y FR, esto no implica que el sistema deba trabajar en ese punto. Para conseguir una buena comparación entre sistemas se suele emplear la representación en forma de curvas DET (Detection Error Tradeoff), que consiste en la presentación de un error frente al otro en un eje normalizado, obteniéndose así una única curva para ambos tipos de error definida por todos los posibles puntos de trabajo del sistema. En esta curva cualquier punto está dado por un valor de FA y otro de FR, de modo que no es necesario estar manejando varias curvas para determinar el punto de trabajo. Por contra, perdemos facilidad para encontrar el EER, que se localiza en la bisectriz del ángulo formado por la parte positiva del eje de FA y FR (Figura 6.3).

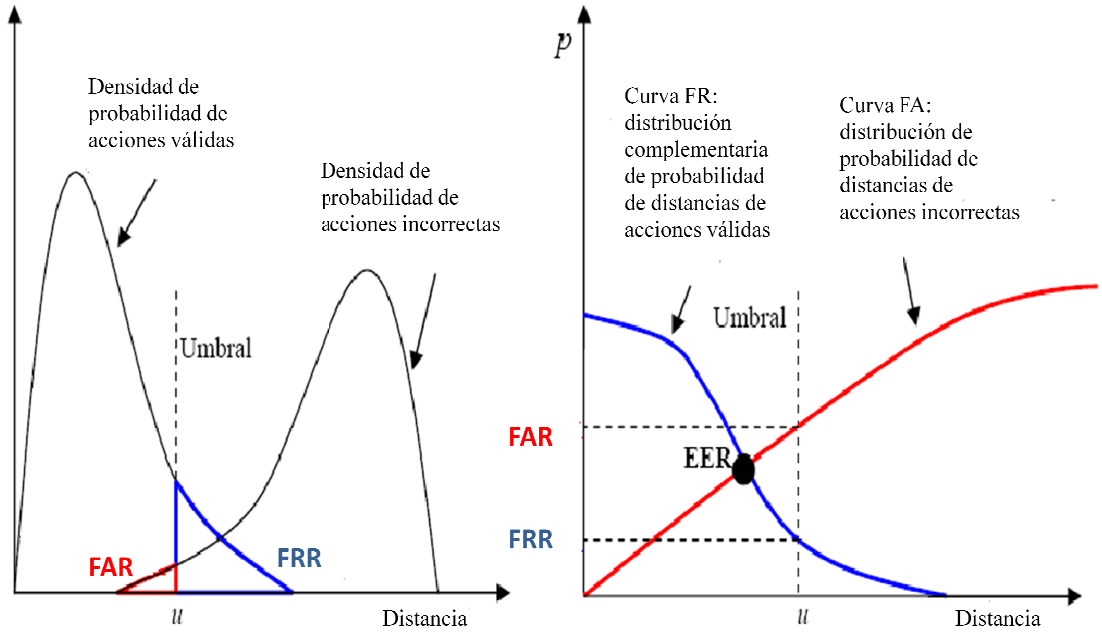


Figura 6.2: Ejemplo de densidades y distribuciones de probabilidad de acciones correctas e incorrectas.

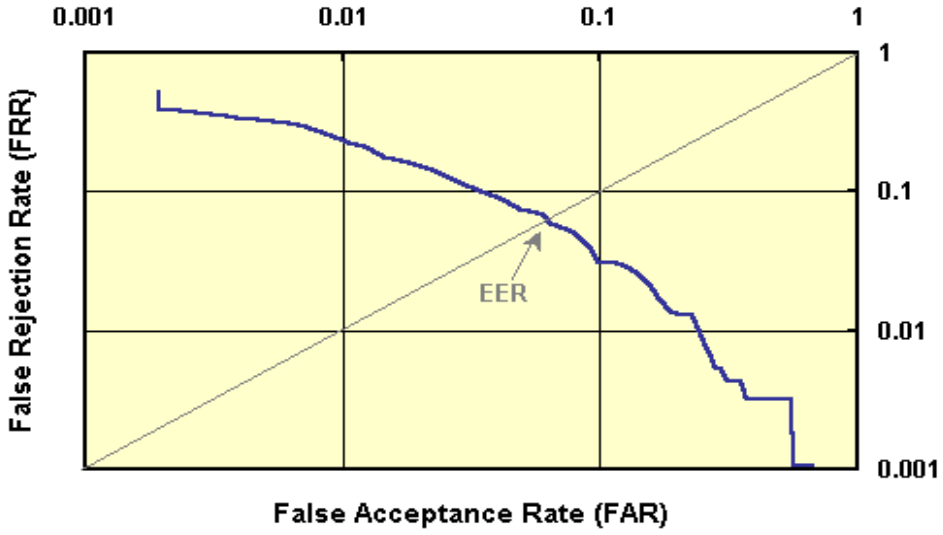


Figura 6.3: Ejemplo de curva DET.

6.1.2. Establecimiento del umbral

Los valores del umbral influyen de forma directa en las tasas de falsa aceptación y falso rechazo. Si el umbral es alto, pocos movimientos válidos serán rechazados, pero un número mayor de movimientos incorrectos serán aceptados de forma errónea, por el contrario, si disminuimos el valor del umbral, decrecerán las falsas aceptaciones a costa de incrementar los falsos rechazos. Por tanto, el establecimiento de umbrales estará condicionado a una especificación de un punto de trabajo que guarde un compromiso entre ambos tipos de error.

Existen dos procedimientos clásicos para establecer los umbrales, ya se realice este proceso a priori o a posteriori.

> ***Establecimiento de umbrales a priori:*** El umbral se establece a partir de un conjunto de datos de estimación, que pueden ser: bien los propios datos de entrenamiento del sistema, o bien un conjunto nuevo de datos no observados hasta el momento. Una vez establecidos los umbrales, las tasas de falso rechazo y falsa aceptación, se estiman a partir de un conjunto de prueba distinto del conjunto usado para la estimación del umbral.
>
> ***Establecimiento de umbrales a posteriori:*** El umbral se calcula a partir de los datos del conjunto de prueba. En este caso, las tasas de falso rechazo y falsa aceptación, deben ser interpretadas como los mejores resultados posibles del sistema, o lo que es lo mismo, el funcionamiento del sistema con un umbral ideal.

Debido a que no existe un estándar a la hora de establecer un umbral en sistemas de reconocimiento de actividades, para concretar el umbral de decisión en nuestro algoritmo hemos optado por ayudarnos de los resultados de las tasas FA y FR. Haciendo un barrido por todos los valores posibles de umbral obtenemos diferentes porcentajes de falsos aciertos y falsos rechazos y según estos resultados determinamos el valor más óptimo para el umbral de decisión en cada caso.

6.2. Resultados

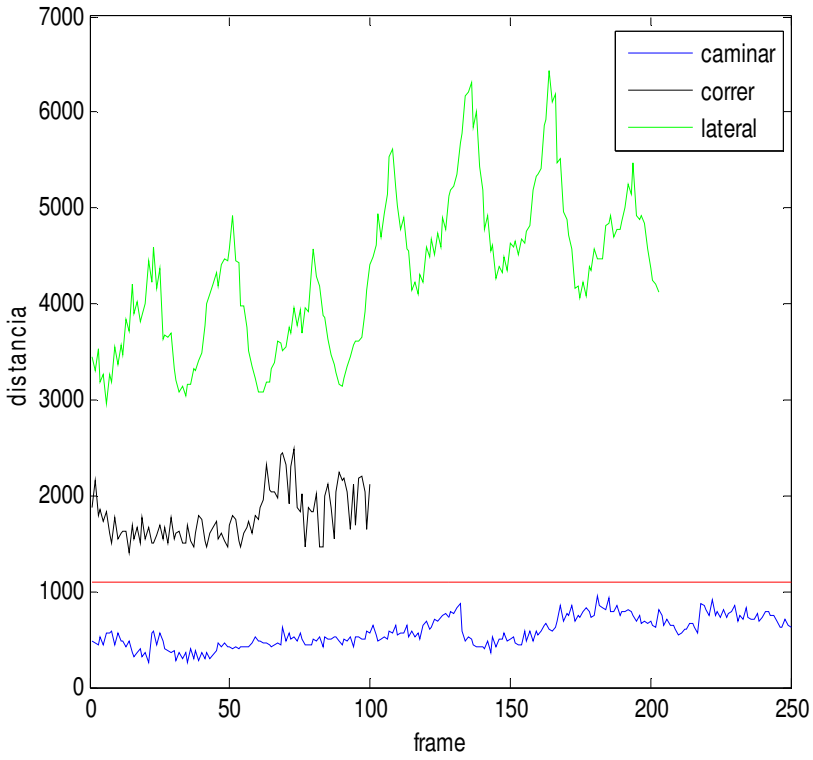
Para realizar los experimentos expuestos a continuación hemos contado con un conjunto de vídeos en los que se muestran diversas actividades humanas que han sido proporcionados por Ivana Mikic, también expondremos posteriormente experimentos realizados sobre un set de vídeos creados exclusivamente para este proyecto (Anexo B).

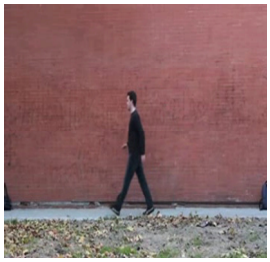
6.2.1. Experimento A: condiciones idénticas

Empezaremos desarrollando el caso más simple que presenta menos dificultades a la hora de reconocer las actividades que aparecen en las secuencias que vamos a analizar. Para este caso las condiciones generales de la secuencia ejemplo y de la secuencia estudiada serán idénticas. Esto implica que el fondo además de ser el mismo para ambas secuencias, será unimodal y los valores de sus pixeles no variaran, tampoco habrá cambios significativos en la iluminación ni en las condiciones de cámara la cual permanecerá en todo momento estática.

A partir de este punto los resultados inmediatos de distancias entre los distintos movimientos, los representaremos mediante una serie de gráficas que relacionaran cada frame en los que aparece una determinada actividad con su correspondiente valor de distancia conductual con respecto de la actividad ejemplo que esta almacenada en nuestra base de datos.

A continuación expondremos los resultados obtenidos, bajo las condiciones descritas anteriormente, de los movimientos caminar, correr y pasos laterales. Primero emplearemos como acción ejemplo caminar (Figura 6.4) y seguidamente utilizaremos como acción ejemplo correr (Figura 6.5).

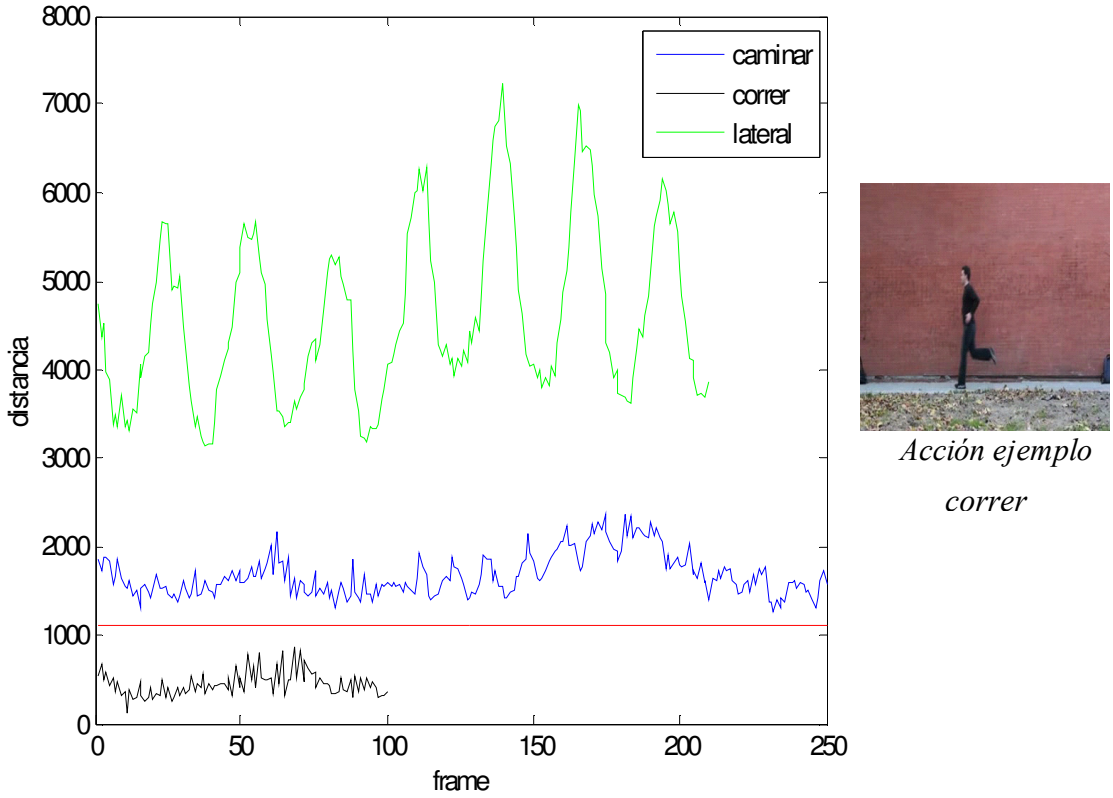




Acción ejemplo caminar

(a) (b) (c)

Figura 6.4: Gráfica con los resultados en el experimento A de las medidas de distancia conductual de las acciones a) caminar, b) correr y c) paso lateral frente a la actividad ejemplo ***caminar****.*



Acción ejemplo correr

(a) (b) (c)

Figura 6.5: Gráfica con los resultados en el experimento A de las medidas de distancia conductual de las acciones a) caminar, b) correr y c) paso lateral frente a la actividad ejemplo ***correr****.*

De los resultados que conseguimos en este experimento, en el que las condiciones son ideales, podemos obtener como conclusión que el sistema consigue discernir con una tasa de error de cero o muy cercana a cero los posibles movimientos que aparecen en la secuencia de vídeo estudiada. Los resultados de las tasas FA y FR se exponen en las tablas siguientes.

	Umbral / Tasas	950	960-1410	1420
Caminar	FA(%)	0	0	0
	FR(%)	0.4000	0	0
Correr	FA(%)	0	0	1.0000
	FR(%)	0	0	0
Pasos laterales	FA(%)	0	0	0
	FR(%)	0	0	0

Tabla 6.1: Tasas FA y FR obtenidas en el experimento A de las acciones caminar, correr y pasos laterales con respecto de la actividad ejemplo ***caminar****.*

	Umbral / Tasas	880	890-1270	1280
Caminar	FA(%)	0	0	0.4000
	FR(%)	0	0	0
Correr	FA(%)	0	0	0
	FR(%)	1.0000	0	0
Pasos laterales	FA(%)	0	0	0
	FR(%)	0	0	0

Tabla 6.2: Tasas FA y FR obtenidas en el experimento A de las acciones caminar, correr y pasos laterales con respecto de la actividad ejemplo ***correr****.*

Conforme a estos resultados hemos optado por poner el umbral de decisión para ambos ejemplos en 1100, ya que en este caso se da que en este punto se puede conseguir perfectamente que las tasas de falsa aceptación y falso rechazo sean cero.

Para realizar esta primera prueba hemos escogido movimientos que son parecidos conductualmente, ya que son este tipo de movimientos los que nos ofrecen una mayor dificultad a la hora de diferenciarlos, los movimientos escogidos para este experimento son los antes mencionados caminar, correr y hacer pasos laterales.

En ambas gráficas (Figuras 6.4 y 6.5) podemos observar que las medidas de distancia que surgen al comparar los pasos laterales con las actividades ejemplo de caminar o

correr se separan de forma significativa del umbral propuesto, lo que nos indica que no tendríamos ninguna dificultad en diferenciar este tipo acción de las acciones ejemplo dadas. Esto también nos lleva a pensar que acciones que se diferencian en su realización aun más que esta acción estudiada (por ejemplo gatear), nos darán como resultado un conjunto de valores de distancia que estarán todavía más alejados del umbral.

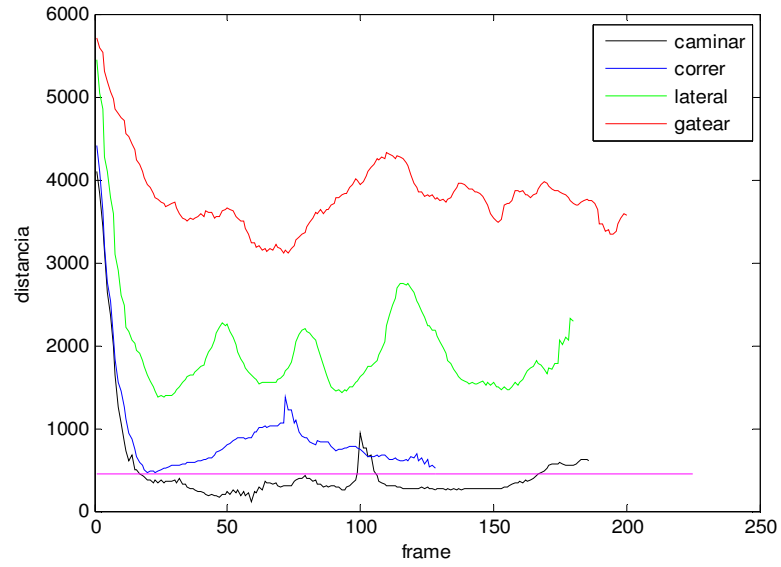
Un caso distinto y más interesante se da a la hora de analizar las actividades de caminar y correr, al ser dos movimientos significativamente parecidos al compararlos entre si obtenemos medidas de distancia relativamente cercanas, lo que hace que a la hora de distinguir una acción de otra se pueda cometer algún tipo de error. En este caso podemos conseguir que el error sea nulo debido a que las condiciones sobre las que trabajamos están muy controladas. Como veremos en los siguientes experimentos, estos dos movimientos serán los que nos ofrecerán una mayor dificultad a la hora de discernir uno del otro, a causa de posibles fenómenos como por ejemplo: oclusiones(zonas en las que se produce efecto camuflaje debido al poco contraste entre el frente y el fondo), cambios significativos de iluminación, etc.

6.2.2. Experimento B: condiciones no idénticas

En este apartado hablaremos de la segunda posibilidad que nos podemos encontrar a la hora de utilizar este sistema. En esta ocasión las condiciones que se dan en el ejemplo no son idénticas a las de las secuencias de vídeo que posteriormente analizaremos, sino que esta vez las condiciones serán parecidas, ya que el ejemplo lo extraeremos de un vídeo en el que el fondo será similar al de los vídeos estudiados, pero en el que puedan aparecer algunas diferencias. Este caso lo podemos calificar como el más habitual, debido a que si aplicásemos este algoritmo a un sistema en la vida real serían estas circunstancias las que se darían con más frecuencia, ya que lo más lógico sería entrenar el sistema con ejemplos sacados de las situaciones con las que nos vamos a encontrar.

Este caso recoge situaciones como: ampliar o disminuir el plano de la cámara, condiciones de iluminación variables, (por ejemplo porque estuviese grabado en exteriores) escenarios con varias cámaras que abarcan toda una zona. La única condición que permanecerá constante siempre es que la cámara debe mantenerse estática.

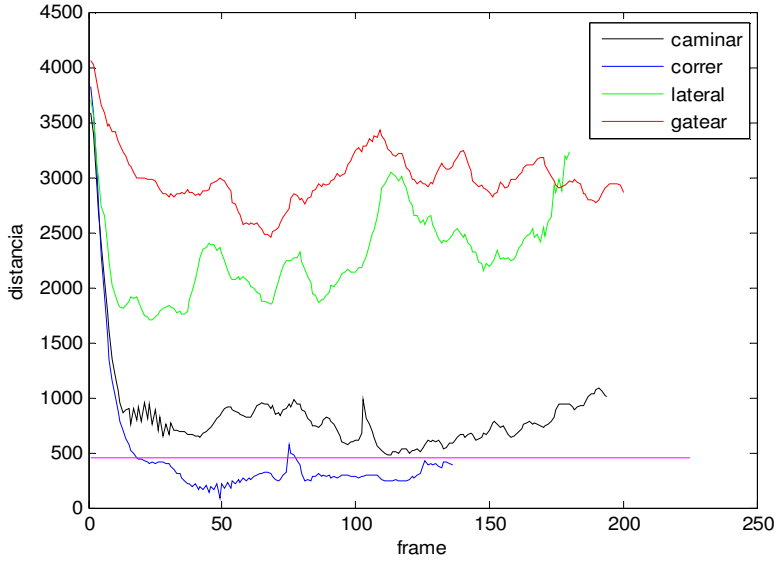
Las siguientes figuras muestran los resultados obtenidos para este experimento de las acciones caminar, correr, paso lateral y gatear. Seguiremos el mismo esquema que el apartado anterior, analizando primero la actividad caminar como ejemplo y luego la actividad correr.



Acción ejemplo caminar

(a) (b) (c) (d)

*Figura 6.6: Gráfica con los resultados en el experimento B de las medidas de distancia conductual de las acciones a) caminar, b) correr, c) paso lateral y d) gatear frente a la actividad ejemplo **caminar**.*



Acción ejemplo correr

(a) (b) (c) (d)

*Figura 6.7: Gráfica con los resultados en el experimento B de las medidas de distancia conductual de las acciones a) caminar, b) correr, c) paso lateral y d) gatear frente a la actividad ejemplo **correr**.*

A simple vista observamos que los resultados no son tan óptimos como en el experimento anterior, aunque las tasas de falsa aceptación y faso rechazo siguen siendo suficientemente bajas como para poder identificar los movimientos que puedan ocurrir en la secuencia. Para tener más datos veamos las tablas que contienen los resultados de las tasas de falsa aceptación y falso rechazo.

	Umbral / Tasas	400	445-463	464-473	500
Caminar	FA(%)	0	0	0	0
	FR(%)	26.8817	22.5806	21.5054	20.4301
Correr	FA(%)	0	0	0.7813	5.4688
	FR(%)	0	0	0	0
Pasos laterales	FA(%)	0	0	0	0
	FR(%)	0	0	0	0
Gatear	FA(%)	0	0	0	0
	FR(%)	0	0	0	0

Tabla 6.3: Tasas FA y FR obtenidas en el experimento B de las acciones caminar, correr, pasos laterales y gatear con respecto de la actividad ejemplo ***caminar****.*

	Umbral / Tasas	400	460-465	476-482	500
Caminar	FA(%)	0	0	0.5155	2.5773
	FR(%)	0	0	0	0
Correr	FA(%)	0	0	0	0
	FR(%)	26.4706	14.7059	14.4861	12.5000
Pasos laterales	FA(%)	0	0	0	0
	FR(%)	0	0	0	0
Gatear	FA(%)	0	0	0	0
	FR(%)	0	0	0	0

Tabla 6.4: Tasas FA y FR obtenidas en el experimento B de las acciones caminar, correr, pasos laterales y gatear con respecto de la actividad ejemplo ***correr****.*

Siguiendo el resultado mostrado en estas tablas hemos decidido colocar el umbral de decisión para el primer ejemplo en 450 y para el segundo ejemplo en 460, ya que en estos dos puntos conseguimos un buen equilibrio entre las tasas de falso rechazo y falsa aceptación respectivamente. El hecho de que la tasa de falso rechazo sea tan elevada se debe a los altos valores de las distancias que aparecen en la parte izquierda de las gráficas, más adelante se explicará el porqué de dicha causa.

El primer movimiento que vamos a analizar es el de gatear, como podemos ver, una vez hemos recopilado todos los datos, es el movimiento cuya distancias conductuales se alejan más de las acciones ejemplo propuestas, lo que nos lleva a concluir que nuestro algoritmo trabaja perfectamente en este tipo de casos, ya que cuanto mayor sean las diferencias a la hora de realizar un movimiento con respecto a la actividad ejemplo mayor es la distancia que obtenemos como resultado. Por lo tanto a la hora de distinguir dos acciones muy diferentes la tasa de error que conseguimos es nula, debido a que las medidas de distancia que resultan están muy separadas del umbral de decisión.

Por otro lado cuando estudiamos el movimiento de pasos laterales con respecto de las actividades ejemplo caminar y correr, podemos sacar unas conclusiones parecidas a las que hemos deducido del movimiento de gatear, porque también este movimiento dista en su realización de las acciones ejemplo, aunque podemos incluir algunos matices. Inicialmente observamos que los valores del conjunto de distancias conductuales son suficientemente altos como para conseguir también unas tasas de falso rechazo y falsa aceptación nulas, por lo que conseguimos diferenciar este movimiento de los dos ejemplos sin cometer errores. De los resultados de este movimiento destacaremos que los valores de distancias que conseguimos son menores dependiendo de si el movimiento se parece más o menos al ejemplo, si comparamos las dos gráficas (Figura 6.6 y 6.7) vemos que las medidas de distancia con respecto a caminar son menores que con respecto a correr, esto se debe a que la acción de pasos laterales más correlacionada con la acción caminar que con la acción correr.

Por último analizaremos los resultados que hemos adquirido de los movimientos de caminar y correr, como sucedía en el experimento anterior, estos dos movimientos al ser los que más se asemejan el uno del otro, son los que al compararlos obtenemos como resultado unas distancias conductuales próximas, esto nos hace más complicado la tarea de identificar estas dos acciones. En este experimento, a causa de que no controlamos todas las condiciones de la escena, no podemos evitar que se produzca algún tipo de error a la hora de comparar estas dos acciones. Algunos errores son inevitables a priori, porque se producen en zonas donde hay efecto camuflaje u oclusiones, pero otros se deben simplemente a que la cercanía entre dos acciones, haciendo que el sistema no pueda conseguir diferenciarlas.

A continuación analizaremos la procedencia de dichos errores y porque se producen, haciendo uso de gráficas como las mostradas en la figura 6.8.

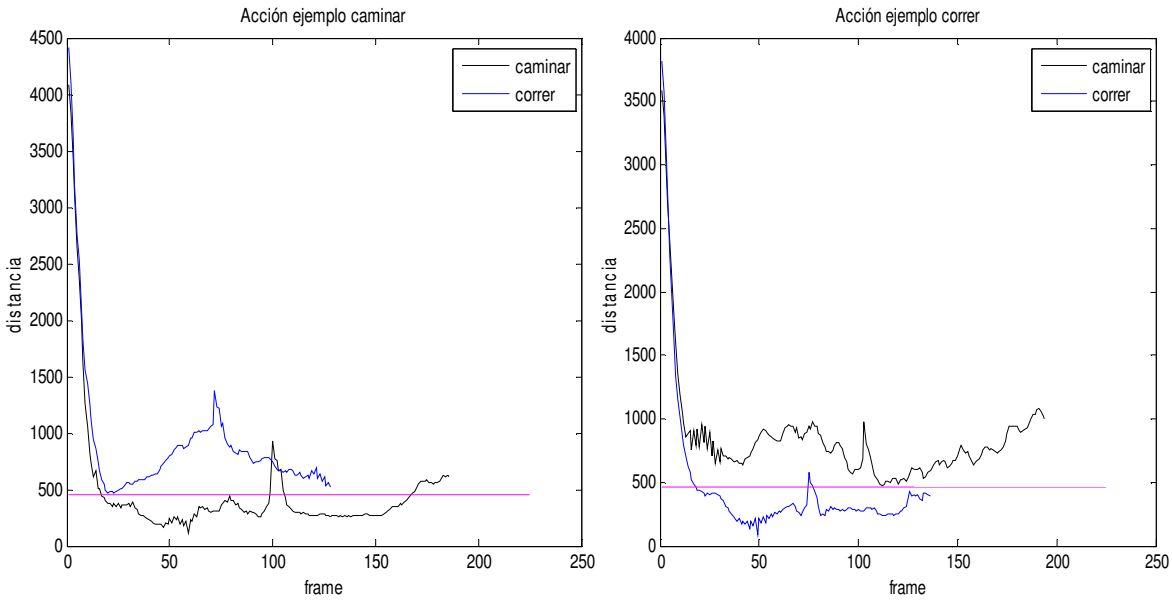
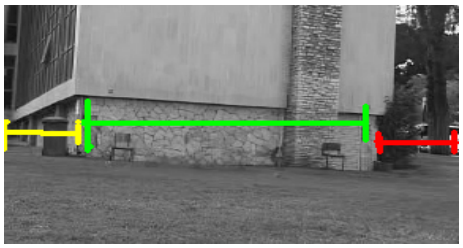


Figura 6.8: Gráficas con los resultados en el experimento B de las medidas de distancia conductual de las acciones caminar y correr con respecto a las actividad ejemplo caminar y correr.



lo en el
s zonas
nta al

Primero aclararemos que los resultados de las dos gráficas (Fig. 6.8) están ordenados de tal forma que la primera mitad se corresponde con el movimiento que transcurre desde la parte derecha hacia la parte izquierda de la pantalla y la segunda mitad el movimiento transcurre de forma opuesta.

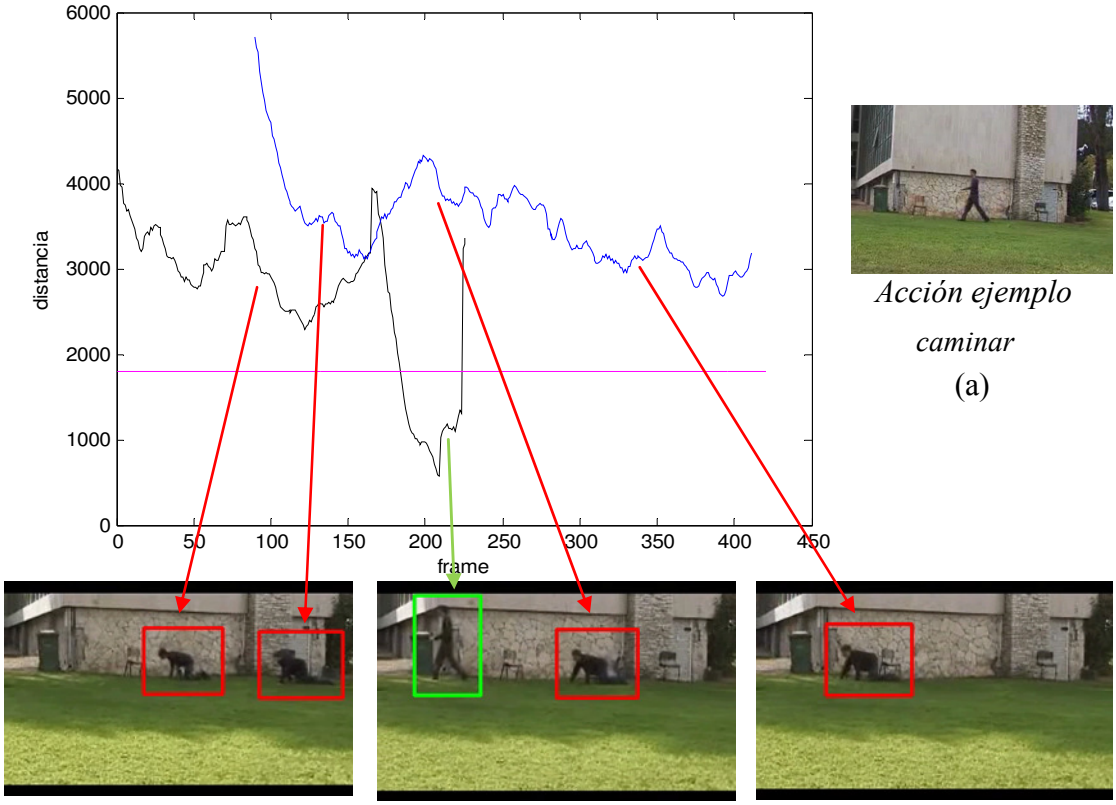
Empezaremos explicando la razón por la que obtenemos unas medidas de distancia tan altas sobre todo en la parte izquierda de las dos gráficas (Figura 6.8) y el porqué también encontramos una pequeña elevación de los valores para ambos movimientos en la parte final. Estos valores altos de distancias se corresponden con la zona del fondo que hemos marcado en rojo (Figura 6.9), en esta zona, como es tan oscura, se produce efecto camuflaje, lo que hace que se pierda de forma casi total la información que necesitamos para caracterizar el movimiento. Este error es mucho más acentuado en la parte izquierda de las gráficas, debido a que cuando un movimiento se inicia específicamente en esa zona no tenemos información previa del mismo, por lo que es mucho más difícil enmendar dicho error.

La zona marcada en amarillo también es un punto de conflicto aunque en menor medida, los resultados que corresponden a esta zona se pueden observar sobre todo en la

parte central de las gráficas (Figura. 6.8), donde se produce claramente un pico en los valores de distancia. En esta zona también se da el efecto camuflaje, pero en esta ocasión sólo afecta a la parte inferior del cuerpo (extremidades inferiores), esto hace que perdamos información de forma parcial. Este error es significativo para las acciones de andar y correr, ya que para estas actividades extraemos bastante información de esa zona del cuerpo. Como en el caso anterior, el error es más acusado cuando interactuamos en esa zona al inicio del movimiento, aunque también se puede apreciar una subida en lo valores, en la zona justo anterior al pico del centro de las curvas, que correspondería a la finalización del movimiento de izquierda a derecha.

En el resto del fondo, marcado en verde, no tenemos ningún problema a la hora de poder diferenciar dos acciones, ya que en esta zona podemos afirmar que el error es casi nulo.

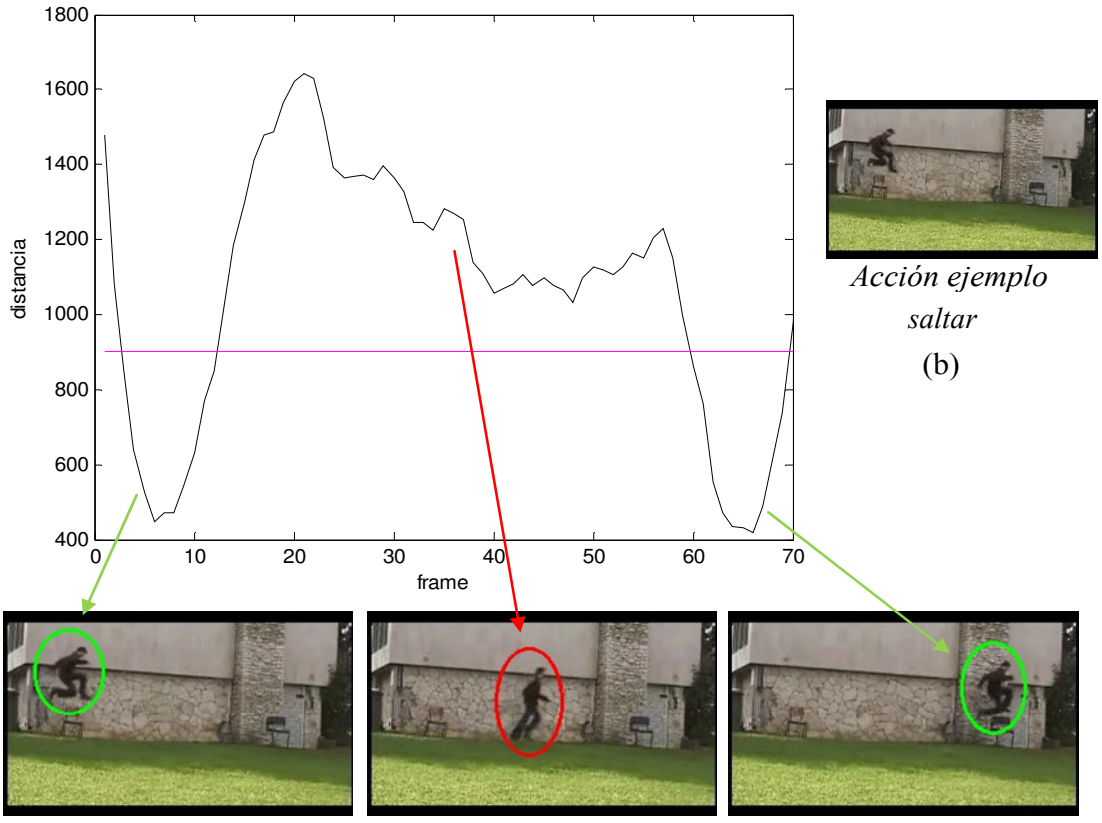
Seguidamente mostraremos algunos datos recogidos de otras secuencias de vídeo que hemos analizado, en los que aparecen varias personas en la secuencia o hay movimientos distintos de los que ya nos hemos referido. Esto nos aportará un mayor conocimiento de las capacidades de nuestro algoritmo.



Acción ejemplo

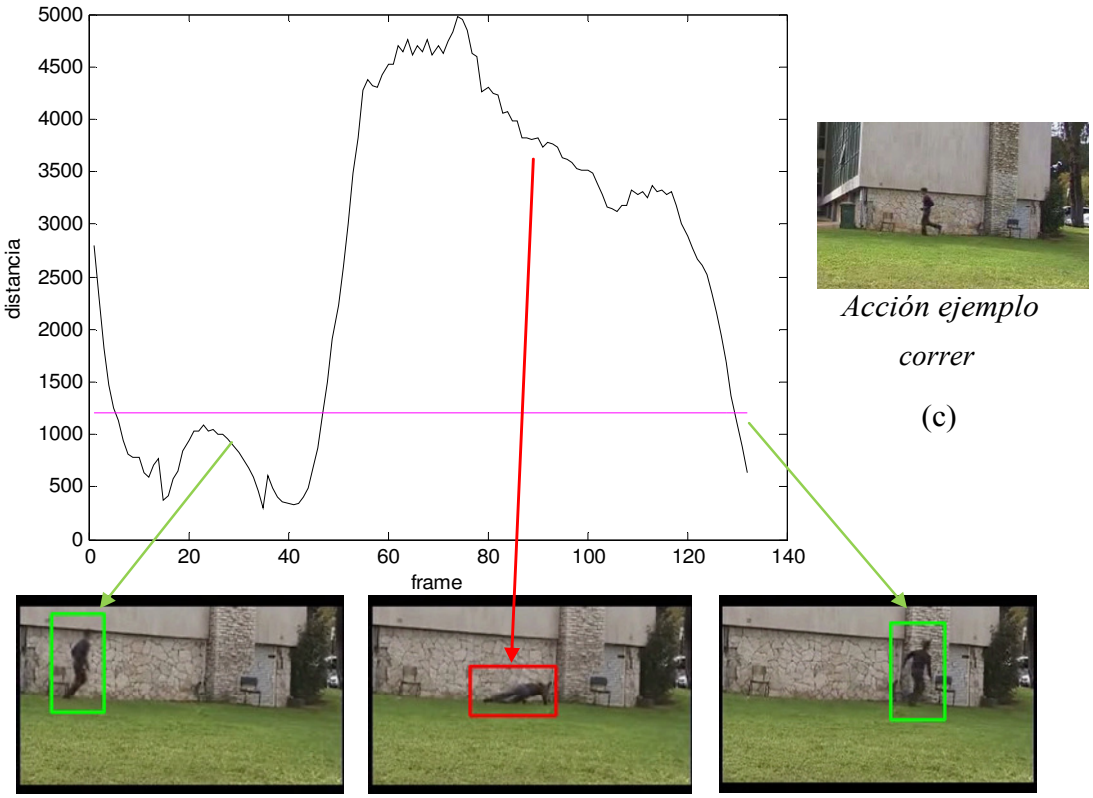
caminar

(a)



Acción ejemplo saltar

(b)



Acción ejemplo correr

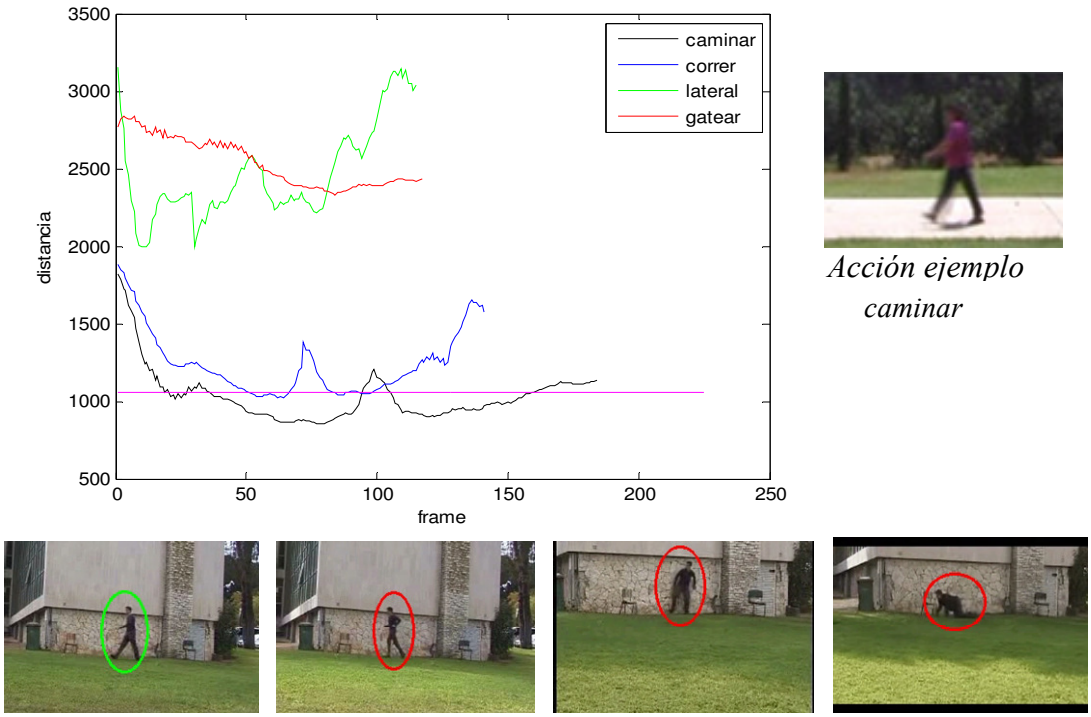
(c)

Figura 6.10: (a), (b) y (c) Gráficas con los resultados en el experimento B de las medidas de distancia conductual de las acciones que encontramos en distintas secuencias de vídeo con respecto a diferentes actividades ejemplo.

6.2.2. Experimento C: condiciones independientes

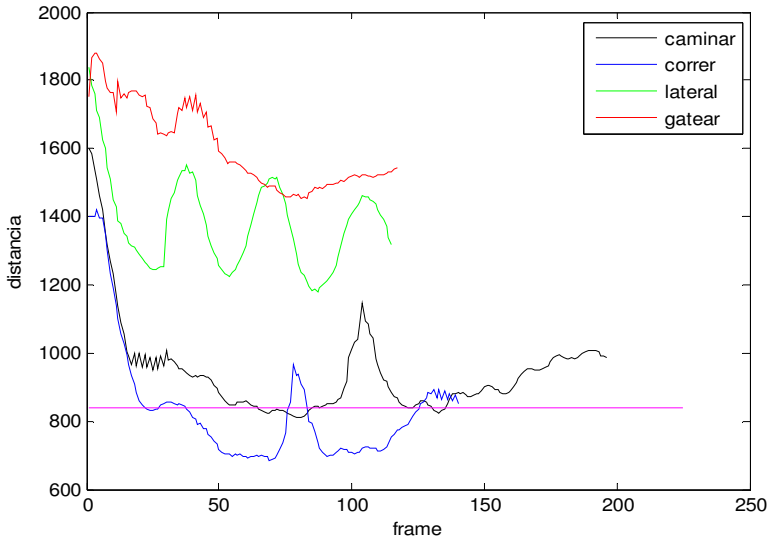
Por último explicaremos el caso más complejo con el que podemos trabajar en este sistema, que consiste en que las condiciones de la secuencia ejemplo y la secuencia que queremos analizar son totalmente independientes, es decir, la acción ejemplo la extraeremos de un vídeo que no tendrá relación con los vídeos con los que trabajaremos posteriormente, por lo que en esta ocasión el fondo, la iluminación y el resto de características de la escena serán distintas. La única condición que sigue permaneciendo constante es que la cámara debe mantenerse estática.

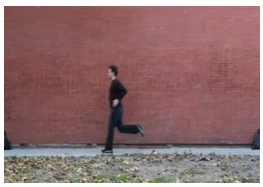
Este modo de trabajar es una de las opciones destacables de este algoritmo, ya que nos permite reconocer las actividades de una secuencia de vídeo usando una actividad ejemplo estándar, lo que nos ahorraría tener que entrenar cada sistema al que aplicásemos nuestro algoritmo. El hecho de que podamos reconocer acciones de esta manera se debe a la forma con la que recogemos las características de los movimientos ayudándonos de los gradientes (x, y, t), con lo que conseguimos eliminar la mayor parte de información que nos puede aportar el fondo de la escena, pudiendo discriminar de esta forma la información que nos aporta el movimiento en cuestión.



Acción ejemplo caminar

Figura 6.11: Gráfica con los resultados en el experimento C de las medidas de distancia conductual de las acciones caminar, correr, paso lateral y gatear frente a la actividad ejemplo ***caminar****.*





Acción ejemplo correr

Figura 6.12: Gráfica con los resultados en el experimento C de las medidas de distancia conductual de las acciones caminar, correr, paso lateral y gatear frente a la actividad ejemplo ***correr****.*

Como hemos representado en los anteriores experimentos, la figuras 6.11 y 6.12 muestran los resultados de los movimientos caminar, correr, pasos laterales y gatear que hemos conseguido bajo las condiciones de este experimento. Inicialmente hemos usado como acción ejemplo caminar y a continuación correr, esta elección se debe a que son los más representativos en la vida real.

Según estos datos es evidente que el error que se comente en este experimento es mayor que en los dos experimentos anteriores, sin embargo aun teniendo más dificultades seguimos pudiendo detectar el movimiento que nos interesa dentro de una escena.

Seguidamente expondremos las tablas con los datos de las tasas de falsa aceptación y falso rechazo.

	Umbral / Tasas	1024	1031	1053	1084	1150
Caminar	FA(%)	0	0	0	0	0
	FR(%)	44.0217	41.3043	35.8696	27.7174	9.2391
Correr	FA(%)	0	2.1277	15.6028	27.6596	41.8440
	FR(%)	0	0	0	0	0
Pasos laterales	FA(%)	0	0	0	0	0
	FR(%)	0	0	0	0	0
Gatear	FA(%)	0	0	0	0	0
	FR(%)	0	0	0	0	0

Tabla 6.5: Tasas FA y FR obtenidas en el experimento C de las acciones caminar, correr, pasos laterales y gatear con respecto de la actividad ejemplo ***caminar****.*

	Umbral / Tasas	810	815	840	856	950
Caminar	FA(%)	0	2.0408	13.2653	28.5714	60.2041
	FR(%)	0	0	0	0	0
Correr	FA(%)	0	0	0	0	0
	FR(%)	47.1429	45.7143	37.1429	28.5714	12.1429
Pasos laterales	FA(%)	0	0	0	0	0
	FR(%)	0	0	0	0	0
Gatear	FA(%)	0	0	0	0	0
	FR(%)	0	0	0	0	0

Tabla 6.6: Tasas FA y FR obtenidas en el experimento C de las acciones caminar, correr, pasos laterales y gatear con respecto de la actividad ejemplo ***correr****.*

Conforme a los resultados de estas tablas hemos colocado el umbral de decisión para el primer ejemplo en 1053 y para el segundo ejemplo en 840, pese a que los puntos ERR para el primer y el segundo ejemplo están en 1084 y 856 respectivamente, no los hemos escogido como umbral debido a que buena parte del porcentaje de error por falso rechazo es provocado por las zonas en las que se dan oclusiones o efecto camuflaje, que no podemos evitar, por lo que preferimos no subir el umbral para no aumentar las falsas aceptaciones de forma innecesaria.

Al igual que sucede en los experimentos previos con los movimientos gatear y pasos laterales, obtenemos un conjunto de distancias conductuales que se alejan lo suficiente del umbral de decisión como para que tanto la tasa de falsos rechazos como la tasa de falsas aceptaciones sean nulas, esto nos lleva a que siempre consigamos distinguir ambos movimientos de acciones como caminar y correr. Las altas distancias

conseguidas nos indican que estas acciones se diferencian bastante en la forma de ejecutarse de las actividades modelo que buscamos. También se da el hecho de que la acción de gatear, al ser la menos parecida a las actividades ejemplo, normalmente resulta que obtiene puntuaciones de distancia superiores a las obtenidas con el movimiento de pasos laterales.

Finalmente pasamos a estudiar los resultados de distancias conductuales para los movimientos de caminar y correr, al ser dos movimientos similares, las curvas que representan las medidas de distancia de estas dos acciones son bastante cercanas, lo que provoca que en algunas ocasiones el algoritmo pueda cometer errores a la hora de intentar distinguir ambos movimientos. En este caso como las condiciones de la secuencia de vídeo que contiene la acción ejemplo son independientes y diferentes de las del vídeo analizado, los errores que comente nuestro sistema son más acentuados. Las zonas del vídeo analizado en las que se produzcan oclusiones o que se dé el efecto camuflaje siguen ocasionando un error inevitable, ya que se pierde información necesaria para la correcta comparación de movimientos. En este experimento por el contrario se comenten un número superior de errores en zonas de la escena en las que no sufrimos efectos negativos, esto puede provocar que en determinadas ocasiones el algoritmo no consiga discernir la acción caminar de la acción correr y viceversa.

6.3. Evaluación de la robustez

En este apartado evaluamos la robustez del sistema de reconocimiento de acción, que presentamos en esta memoria, analizando como afectan a nuestra detección determinados parámetros.

6.3.1. Cambio de la longitud de ventana temporal

Para analizar cada secuencia de vídeo introducida en el sistema, como ya hemos explicado en secciones anteriores, vamos dividiendo la secuencia en subsecuencicas de longitud ‘M’ frames, de estas subsecuencias extraemos la información necesaria para más adelante compararla con la información de la acción ejemplo buscada y tomar la decisión que corresponda. ‘M’ es el número de frames de los que consta la secuencia de nuestra actividad ejemplo.

Escogiendo subsecuencais de tamaño 'M' logramos que tengan la misma longitud que la secuencia que contiene la actividad ejemplo, aunque esto es recomendable para asegurarnos el buen funcionamiento del sistema, no implica que sea estrictamente necesario que la subsecuencia tenga que tener el mismo tamaño que la secuencia del ejemplo o que contenga información de un movimiento en todos y cada uno de los 'M' frames que la componen.

El tamaño de la ventana que escogemos para dividir la secuencia de vídeo analizada puede variar, pudiendo mantener un reconocimiento de calidad de los movimientos que aparecen en el vídeo. Según las pruebas y observaciones realizadas, si la longitud de nuestra ventana temporal, que marca el tamaño de cada subsecuencias, es superior a las tres cuartas partes de 'M' nuestro algoritmo seguirá reconociendo perfectamente las acciones buscadas. Por otro lado si el tamaño de la subsecuencia está comprendido entre 'M/2' frames y '3M/4' frames el algoritmo sigue reconociendo las acciones iguales al ejemplo, pero con mayor dificultad, lo que hace disminuir la calidad de nuestro sistema. En cambio si la longitud de la subsecuenia es inferior a la mitad del número total de los frames de la secuencia ejemplo, podemos decir que no conseguiríamos suficiente información para lograr un reconocimiento fiable.

Esto es aplicable a las zonas de las distintas escenas en las que se producen oclusiones o se da el efecto camuflaje, ya que esto nos hace que perdamos información sobre un movimiento en cierto número de frames de la subsecuencia. Si perdemos información en un número reducido de frames de la subsecuencia nuestro sistema conseguirá detectar de forma óptima la actividad buscada, si por el contrario el número de frames en los que perdemos información es superior a la mitad de 'M' nuestro sistema no lograría en la mayoría de los casos reconocer el movimiento.

6.3.2. Fondo multimodal

Existen dos tipos de fondo los fondos unimodales y los fondos multimodales, la característica principal de los primeros consiste en que los pixeles del fondo a lo largo de toda una secuencia de vídeo permanecen constantes, en cambio en los fondos multimodales los pixeles del fondo pueden sufrir alguna variación a lo largo de la secuencia.

Hasta ahora hemos demostrado la eficacia de nuestro sistema trabajando sobre fondos unimodales, pero este algoritmo también tiene la capacidad de trabajar con secuencias en las que se dé un fondo multimodal, como por ejemplo en escenas en las que el fondo se mueva de forma lenta y periódica (olas del mar, hojas agitándose).

En la figura 6.13 podemos ver algunas imágenes extraídas de pruebas realizadas sobre una secuencia de vídeo grabada en una playa, es decir, una secuencia con un fondo multimodal. La actividad ejemplo que hemos utilizado en esta prueba es caminar.

Figura 6.13: Ejemplo del reconocimiento de acciones con una secuencia con fondo multimodal.

Como se puede ver en las imágenes nuestro sistema consigue reconocer acciones en una secuencia con un fondo en el que los pixeles varían de forma periódica, para ello además necesitamos que la cámara permanezca estática. Sin embrago si los pixeles del fondo variasen constantemente de forma no periódica o la cámara no permaneciese estática nuestro algoritmo fallaría y no podríamos reconocer correctamente las actividades que buscamos.

6.3.3. Cambios en la vestimenta

El sistema es robusto a los cambios en el color de la vestimenta infligidos por diferentes prendas de vestir, esto deja de ser así cuando intervienen en la escena personas con prendas con diferentes texturas, ya que si la vestimenta que lleva la persona que realiza el movimiento que estamos analizando incluye una prenda a rayas, puede provocar que los gradientes en los ejes 'x' o 'y' varíen, esto nos llevaría a introducir información errónea en los histogramas, lo que puede hacer que el reconocimiento de la actividad en cuestión falle.

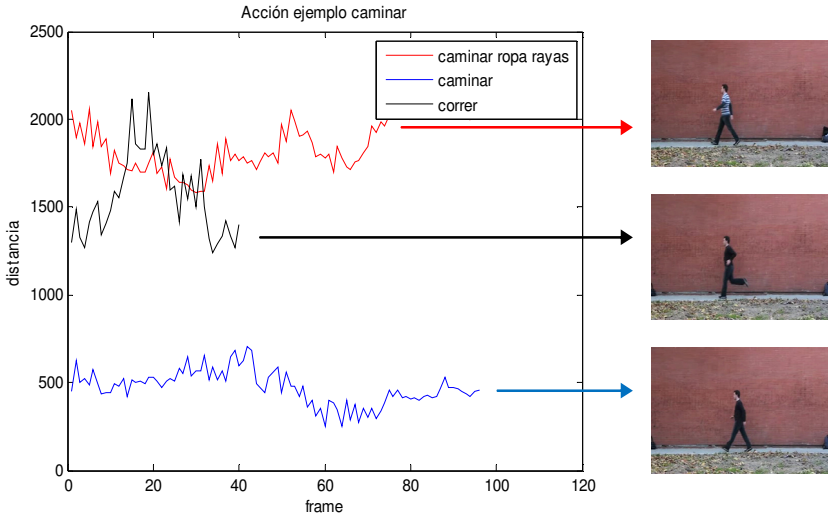


Figura 6.14: Gráfica que muestra las medidas de distancia conductual de una persona caminando y corriendo, con ropa de un color homogéneo, y una persona caminando con una prenda de vestir a rayas frente a la actividad ejemplo caminar.

Mediante los datos que nos aporta la gráfica (Figura 6.14) podemos comprobar que si incluimos una prenda a rayas, en la vestimenta de la persona que realiza el movimiento que estamos estudiando, la distancia que obtenemos es bastante mayor que la que deberíamos conseguir, lo que provoca que la tarea de reconocer la acción que buscamos pueda fallar. En este ejemplo en concreto, la acción de caminar portando una prenda a rayas consigue valores de distancia similares a los de la acción correr.

Para solucionar este contratiempo tendríamos que utilizar un método que nos enturbie y nos mezcle los distintos colores de la prenda rayada, así el sistema tomaría esa prenda como si fuese de un único color y de esta forma conseguiríamos que la prenda no tuviese influencia sobre los gradientes espaciales.

6.3.4. Variación de las escalas temporales

Para realizar este sistema hemos tenido en cuenta que una misma acción humana puede ocurrir a diferentes velocidades, por eso hemos construido nuestro algoritmo para que sea robusto frente a estos cambios de velocidad. Esto lo conseguimos gracias a la utilización de la pirámide temporal que creamos en la primera parte del algoritmo, consiguiendo así replicar los datos del mismo movimiento en distintas escalas temporales, lo que nos resulta de gran ayuda a la hora reconocer un determinado movimiento si este se ejecuta en una escala temporal distinta que la de la actividad ejemplo.

6.3.5. Variación de las escalas espaciales

Las variaciones en el tamaño espacial debido a cambios moderados de zoom o de la distancia de la persona con respecto de la cámara de vídeo provocan únicamente un efecto pequeño sobre el conjunto de gradientes, por lo tanto, podemos asegurar que nuestro sistema maneja correctamente estos pequeños cambios. Por otra parte, si se dan cambios grandes en el zoom o en la distancia de la persona con respecto a la cámara, estos si afectaran de forma más clara a los gradientes, lo que acarreara consecuencias negativas a nuestro reconocedor de actividades humanas.

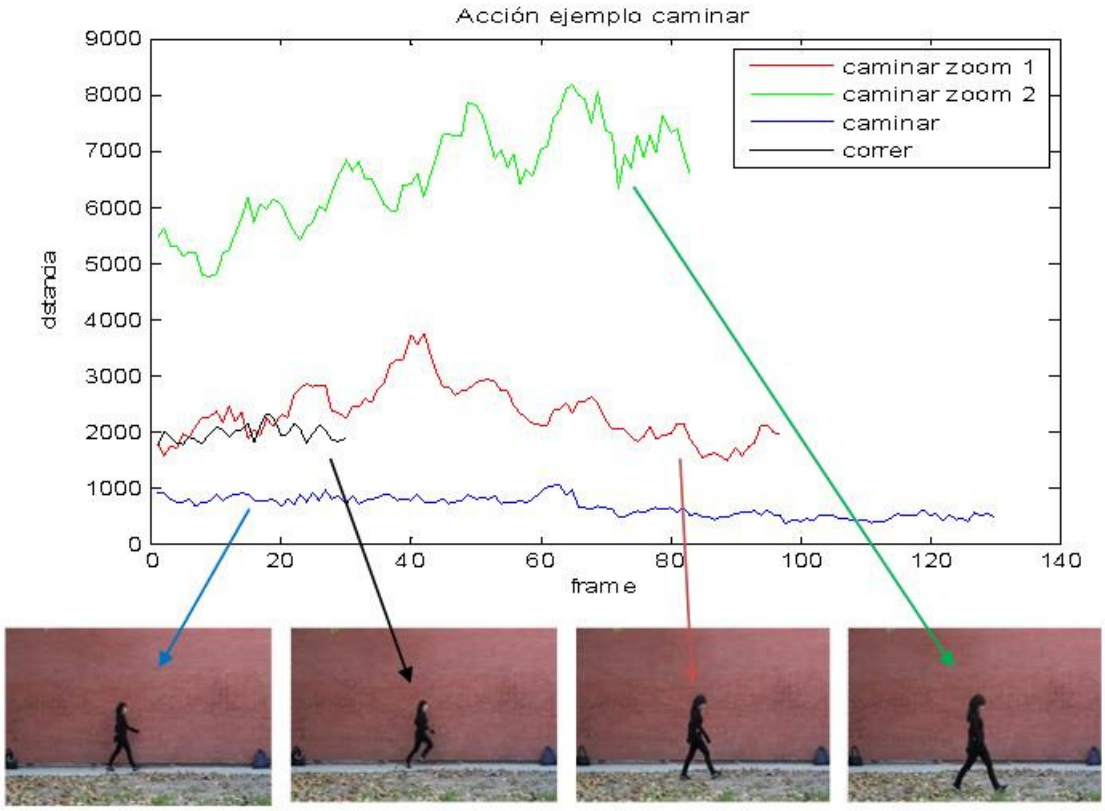


Figura 6.15: Gráfica que muestra las medidas de distancia conductual de una persona caminando a distintas distancias con respecto a la cámara frente a la actividad ejemplo caminar.

Si observamos los resultados de la gráfica (Figura 6.15) podemos corroborar perfectamente, que los cambios significativos de la distancia que existe entre la persona que aparece en la escena y la cámara afectan directamente a los valores de distancia conductual. Podemos concluir, que aunque se realice el mismo movimiento las variaciones de las escalas espaciales hacen que nuestro sistema no consiga identificar correctamente las acciones que pueda contener una secuencia. También podemos ver que cuanto mayor sea el cambio de zoom o el cambio de distancia de la persona con respecto a la cámara, mayor será el efecto sobre la orientación de los gradientes y menor será la posibilidad de reconocimiento.

Para vencer este problema debemos utilizar una solución parecida a la usada para evitar las variaciones de las escalas temporales, tenemos que construir para cada acción que queramos detectar una representación en múltiples escalas espaciales. Para ello podemos introducir varios ejemplos para cada acción que se diferencien únicamente en el zoom utilizado o en la distancia con respecto a la cámara.

6.3.6. Cambios en la dirección de visión

Los cambios significativos en la dirección de la visión de la cámara o cambios en la dirección del movimiento que realiza una persona con respecto a la cámara influyen directamente en el conjunto de gradientes espacio-temporales. Teniendo en cuenta que una persona puede estar realizando un mismo movimiento frente a una cámara de vídeo y en cualquier momento puede cambiar el ángulo de la dirección del movimiento con respecto a la cámara, hemos realizado un estudio para conocer cómo afectan estos cambios al sistema, los resultados se exponen a continuación.

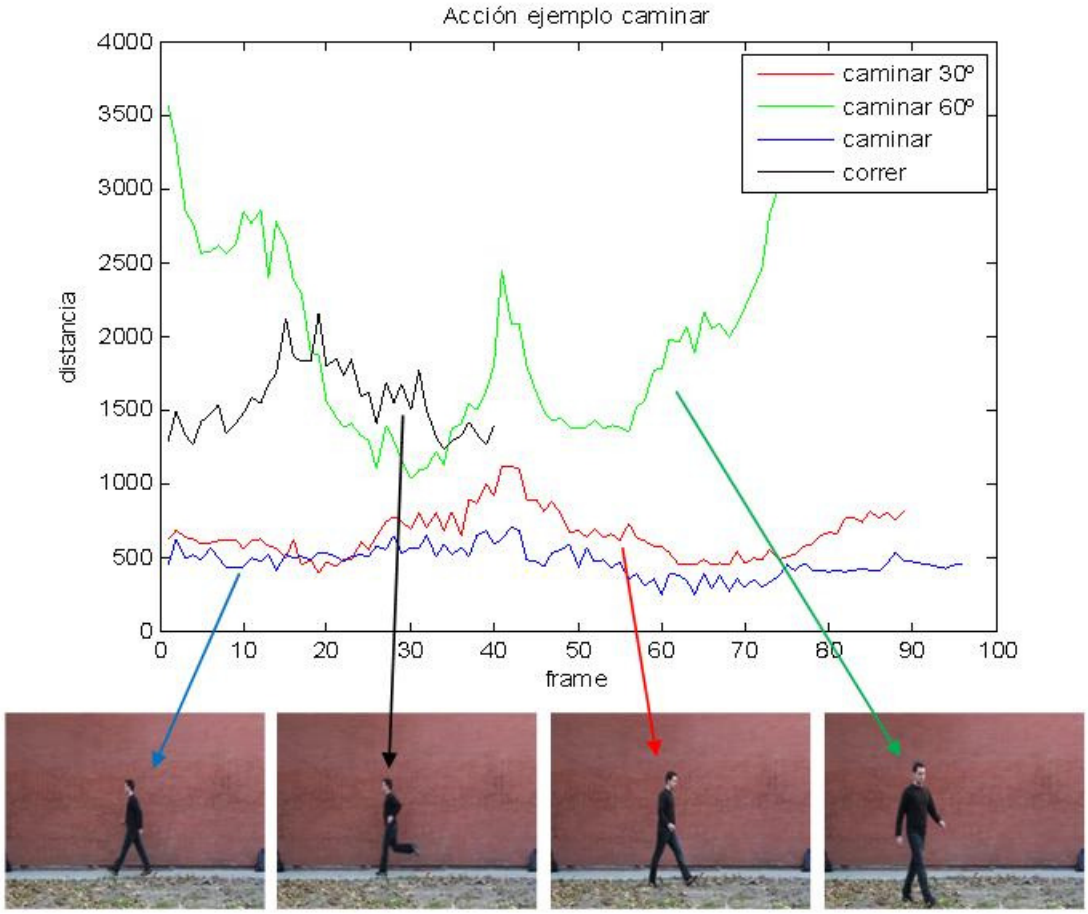


Figura 6.16: Gráfica que muestra las medidas de distancia conductual de una persona caminando con diferentes direcciones de movimiento con respecto a la cámara frente a la actividad ejemplo caminar.

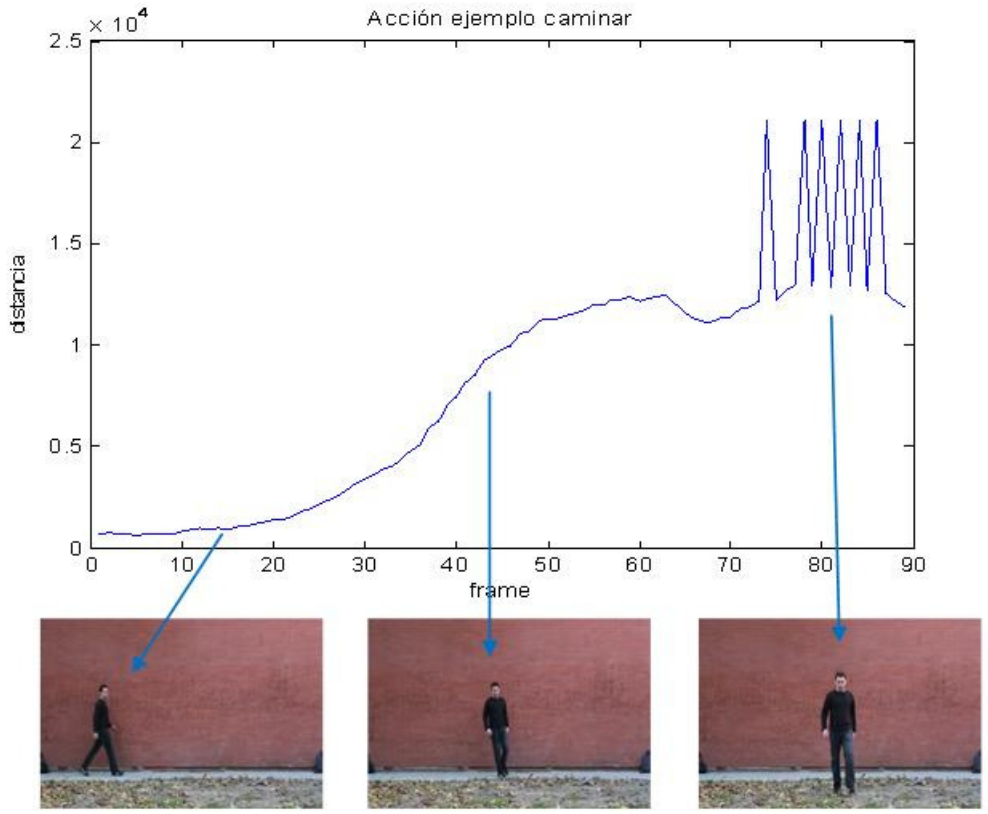


Figura 6.17: Gráfica que muestra las medidas de distancia conductual de una persona caminando, en un primer momento paralelo a la cámara y seguidamente cambiando su dirección de movimiento 90º, comenzando así a caminar perpendicular a la cámara frente a la actividad ejemplo caminar.

Los datos que nos ofrecen estas gráficas (Figura 6.16 y 6.17) nos sirven de gran ayuda para conocer como actúa nuestro sistema a cambios en la dirección de una persona o cambios en la dirección de visión de la cámara. Si observamos las gráficas podemos decir que cuanto mayor sea el cambio de dirección, más afecta a los gradientes y por lo tanto a la tarea de reconocimiento de actividades. Si el cambio de dirección es de 90º con respecto a la dirección del movimiento ejemplo, aunque se trate de la acción que queremos detectar, el sistema asigna a este movimiento unos valores de distancia muy altos por lo que será imposible reconocer esa acción. Por otro lado, si el cambio de dirección es de 60º el efecto se reduce considerablemente, pero los valores de distancia que obtenemos pueden hacer que nos confundamos con otros movimientos, como por ejemplo, como ocurre en este caso, con la acción correr. Sin embargo, vemos que si el cambio de dirección del movimiento con respecto a la cámara es de 30º o menor el sistema no se ve afectado y por lo tanto manejaría esta situación reconociendo correctamente la actividad incluida en el ejemplo.

Para contrarrestar este problema debemos introducir en el sistema, para cada acción buscada, varios ejemplos cuyas direcciones de realización con respecto a la cámara sean distintas.

7. Conclusiones y trabajo futuro

En el presente proyecto se ha estudiado, desarrollado, implementado y documentado un sistema reconocedor de actividades humanas en vídeo, nuestro algoritmo es capaz de detectar y reconocer las diferentes acciones que aparecen en una secuencia de vídeo sirviéndose sólo de una simple medida de distancia estadística, con la que podemos medir la semejanza conductual entre dos movimientos.

Como podemos comprobar, gracias a los resultados obtenidos expuestos en esta memoria, hemos logrado cumplir los objetivos propuestos en un principio para desarrollar un sistema óptimo que consigue manejar múltiples actividades humanas presentes en diferentes secuencias de vídeo, para ello nos hemos basado en el trabajo previo de Zelnik-Manor e Irani descrito en el documento "Statistical Analysis of Dynamic Actions" [1].

El primer objetivo que se ha conseguido es que el sistema sea capaz de trabajar con todo tipo de acciones dinámicas, esto se debe a la forma con la que se extraen las características de los movimientos que surgen en las secuencias con las que trabajamos, ya que solamente nos centramos en escoger los rasgos espacio-temporales en múltiples escalas temporales, conservando así las peculiaridades conductuales de las acciones y desechando cambios en el fondo, iluminación, aspecto, etc. Por otro lado, esto también nos ha ayudado a disminuir considerablemente la duración de la etapa de estudio de las actividades que vamos a tomar como ejemplo con respecto de otros sistemas similares, debido a que para caracterizar las acciones necesitamos un número reducido de parámetros.

El segundo objetivo que hemos logrado satisfactoriamente es simplificar el método mediante el cual se comparan dos actividades que aparecen en dos secuencias de vídeo distintas. Para ello nos ayudamos de una medida de distancia no paramétrica que está basada únicamente en el comportamiento, con esta medida de distancia conductual logramos decidir si las acciones humanas aprendidas anteriormente están o no presentes en el vídeo estudiado.

Concluyendo, el sistema desarrollado: reconocimiento de actividades en vídeo basado en ejemplos, es muy versátil y que está capacitado para poder trabajar con una amplia gama de acciones y vídeos, ya que no es necesario que las condiciones de la secuencia o secuencias que contiene la actividad ejemplo sean iguales o similares a las del vídeo que queremos analizar.

A lo largo de la elaboración de este proyecto hemos podido comprobar que la tarea de reconocimiento de acciones en video es bastante difícil. Actualmente es un aspecto pendiente en el mundo del procesamiento imagen y vídeo. Muchos sistemas están enfocados al reconocimiento de un conjunto altamente limitado de acciones, sin embargo, los sistemas reales tendrán que extenderse en variedad y número de acciones para poder manejarse, es por esto que en este proyecto se intenta seguir esa dirección, aunque este sistema que presentamos aún está muy lejos de ser definitivo. Mejores métodos han ser desarrollados todavía, para conseguir la invariancia ante cambios en la dirección de visión, apariencia, distancia a la cámara, etc.

Referencias

[1] L. Zelnik-Manor and M. Irani, “Statistical analysis of dynamic actions”, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp.1530-1535, 2006.

[2] Y. Tsaig and A. Averbuch, “Automatic Segmentation of Moving Objects in Video Sequences: A Region Labeling Approach”, IEEE Transactions on Circuits and Systems for Video Technology, pp. 597-612, Dec. 2002.

[3] H. Dias, J. Rocha, P. Silva, C. Leao, “Distributed Surveillance System", Conference on Artificial Intelligence, pp 257-261, 2005.

[4] M. Valera and S.A. Velastin, “Intelligent Distributed Surveillance Systems”, IEEE Proc.-Vis. Image Signal Processing., vol. 152, no. 2, Apr. 2005.

[5] M. M. Chang, M. I. Sezan, and A. M. Tekalp, “An algorithm for simultaneous motion estimation and scene segmentation,” IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. V, pp. 221–224, Adelaide, Australia, Apr. 1994.

[6] R.Lienhart, C.Kuhmunch, W. Effelsberg, “On the detection and recognition of television commercials”, International Conference on Multimedia Computing and Systems, pp. 509–516, 1997.

[7] O. Sukmarg, K.R. Rao, “Fast object detection and segmentation in MPEG compressed domain”, IEEE TENCON, vol. 3, pp. 364–368, Kuala Lumpur, Malaysia, 2000.

[8] K. Toyama, J. Krumm, B. Brumitt, B. Meyers, “Principles and Practice of Background Maintenance”, ICCV, Seventh International Conference on Computer Vision, vol.1, pp. 255-261, 1999.

[9] A. Elgammal, R. Duraiswami, D. Harwood, LS. Davis. “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance”, IEEE, pp. 1151-1163 Jul. 2002.

[10] M. Harville, G. Gordon, J. Woodfill “Foreground Segmentation Using Adaptive Mixture Models in Color and Depth”, IEEE Workshop on Detection and Recog of Events in Video, pp. 3-12, 2001.

[11] R Ewerth, B Freisleben, “Frame difference normalization: an approach to reduce error rates of cut detection algorithms for MPEG videos” ICIP, pp. 1009-1012, 2003.

[12] B.P.L. Lo and S.A. Velastin, “Automatic congestion detection system for underground platforms”, International Symposium on Intelligence Multimedia, Video and Speech Processing, pp. 158-161, 2000.

[13] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, “Detecting moving objects, ghosts and shadows in video streams”, IEEE Trans. Patter Analysis and Machine Intelligent, vol. 25, no. 10, pp. 1337-1342, Oct. 2003.

[14] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, “Pfinder:Real-time Tracking of the Human Body,” IEEE Trans. Patter Analysis and Machine Intelligent, vol. 19, no. 7, pp. 780-785, 1997.

[15] C. Stauffer, W.E.L. Grimson, “Adaptive background mixture modelsfor real-time tracking”, CVPR, pp. 246-252, 1999.

[16] A, Elgammal, D. Harwood, and L.S. Davis, “Non-parametric Model for Background Subtraction”, ICCV '99 FRAME-RATE Workshop, 1999.

[17] D. Butler, S. Sridharan, VMJr. Bove, “Real-time Adaptive Background Segmentation. Acoustics, Speech, and Signal Processing”,2003. Proceedings. (ICASSP '03). 2003 IEEE, pp.349-52, Apr. 2003.

[18] A. Cavallaro, T. Ebrahimi, “Change detection based on color edges, circuits and systems”, IEEE International Symposium, pp. 141-144, May, 2001.

[19] L.Fuentes, S.Velastin, “From tracking to advanced surveillance”, IEEE International. Conference on Image Processing, Barcelona, Spain, Sept 2003.

[20] T. Boult, R. Micheals, X. Gao, M. Eckmann, “Into the woods: Visual surveillance of oncooperative camouflaged targets in complex outdoor settings”, IEEE, pp. 1382- 1402, Oct. 2001.

[21] R. J. Schalkoff, “Digital Image Processing and Computer Vision” Editorial John Wiley & Sons pp. 213-218, 1989.

[22] B.K.P. Horn, “Robot Vision”, MIT Press, Cambridge, MA, & McGrawn-Hill Book Company, New York, 1986.

[23] M. Wessler, L.A. Stein, “Robust Active Vision from Simple Symbiotic Subsystems”, Technical Report, MIT, 1997.

[24] P. Anandan, “A computacional cuadrowork and an algorithm for the measurement of visual motion”, International Journal of Computer Vision, pp.283-310, Jan., 1989.

[25] S. A. Brock-Gunn, G.R. Dowling, T. J. Ellis. “Tracking using colour information”, Technical Report TCU/CS/1994/7, City Univ. London, 1994.

[26] M. Kass, A. Witkin and Terzopoulos. “Snakes: Active contour models”, International Journal of Computer Vision, pp-133-144, 1987.

[27] S. Gil, R. Milanese, T. Pun. “Feature selection for object tracking in traffic scenes”, SPIE International Symposium on Smart Highways, Boston, Massachusetts, Oct. 31-Nov. 4, 1994.

[28] J.C. Niebles, F.F. Li, “A Hierarchical Model of Shape and Appearance for Human Action Classification”, CVPR07(1-8), IEEE, 2007.

[29] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. “Actions as space-time shapes”, ICCV, 2005.

[30] C. Fanti, L. Zelnik-Manor, and P. Perona. Hybrid models for human motion recognition. In CVPR, 2005.

[31] N. Dalal, B. Triggs, and C. Schmid. “Human detection usingoriented histograms of flow and appearance”. ECCV, 2006.

[32] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. “Recognizing action at a distance”. In ICCV, 2003.

[33] V. Cheung, B. J. Frey, and N. Jojic. “Video epitomes”. InCVPR, 2005.

[34] J. C. Niebles, Hongcheng Wang and Li Fei-Fei. “Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words”. International Journal of Computer Vision (IJCV). Sep., 2008.

[35] L. Fei-Fei, & P.Perona, “A Bayesian hierarchical model for learning natural scene categories2, IEEE computer society conference on computer vision and pattern recognition, pp. 524–531, 2005.

[36] T. Hofmann, “Probabilistic latent semantic indexing”, 22nd annual international ACM SIGIR conference on research and development in information retrieval pp. 50–57, Aug. 1999.

[37] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation Journal of Machine Learning Research”, vol. 3, pp. 993–1022, 2003.

[38] M. Marsza lek, I. Laptev and C. Schmid. “Actions in context”, IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[39] B. Sch¨olkopf and A. Smola. “Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond”, MIT Press, Cambridge, MA, 2002.

[40] A. A. Efros, A. C. Berg, G. J. Mori, and Malik, “Recognizing action at a distance”. IEEE international conference on computer visión, Vol. 2, pp. 726–733, 2003.

[41] E. Shechtman, and M. Irani, “Space-time behavior based correlation”. IEEE computer society conference on computer vision and pattern recognition, Vol. 1, pp. 405– 412), 2005.

[42] D. Ramanan, D. A. and Forsyth, “Automatic annotation of everyday movements”. In Thrun, S., Saul, L., & Schölkopf, B. (Eds.), Advances in neural information processing systems, Vol. 16, Cambridge: MIT Press, 2004.

[43] A. Yilmaz, and M. Shah, "Recognizing human actions in videos acquired by uncalibrated moving cameras", IEEE international conference on computer vision Vol. 1, pp. 150–157, 2005.

[44] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R.& Basri, "Actions as space-time shapes", IEEE international conference on computer visión, Vol. 2, pp. 1395– 1402, 2005.

[45] H. Zhong, J. Shi, & M. Visontai, "Detecting unusual activity in video". IEEE computer society conference on computer vision and pattern recognition pp. 819–826, 2004.

[46] Y. Ke, R. Sukthankar, & M. Hebert, "Efficient visual event detection using volumetric features". IEEE international conference on computer visión, pp. 166–173, 2005.

[47] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach". ICPR, pp. 32–36, 2004.

[48] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features". IEEE international workshop on visual surveillance and performanc., 2005.

[49] A. Oikonomopoulos, I. Patras, and M. Pantic, "Human action recognition with spatiotemporal salient points", IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, pp. 710–719, 2006.

[50] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering objects and their location in images", IEEE international conference on computer vision pp. 370–377 Oct. 2005.

[51] A. K. Jain, A. Ross, and S. Prabhakar. "An introduction to biometric recognition". IEEE Transactions on Circuits and Systems for Video Technology, pp.125_143, 2006.

[52] A. Bobick and J. Davis, "The Representation and Recognition of Action Using Temporal Templates," IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 23, no. 3, pp. 257-267, Mar. 2001.

[53] G. Bradski and J. Davis, "Motion Segmentation and Pose Recognition with Motion History Gradients," Int'l J. Machine Vision and Applications, vol. 13, no. 3, pp. 174-184, 2002.

[54] O. Chomat and J.L. Crowley, "Probabilistic Recognition of Activity Using Local Appearance," IEEE Conference Computer Vision and Pattern Recognition, June 1999.

[55] A. Efros, A. Berg, G. Mori, and J. Malik, "Recognizing Action at a Distance," Int'l Conf. Computer Vision, vol. 2, pp. 726-733, Oct. 2003.

[56] D.M. Gavrila and L.S. Davis, "3-D Model-Based Tracking of Humans in Action: A Multi-View Approach," Proc. IEEE Confeference Computer Vision and Pattern Recognition, 1996.

[57] M. Irani, B. Rousso, and S. Peleg, “Computing Occluding and Transparent Motions,” Int’l J. Computer Vision, vol. 12, pp. 5-16, Feb. 1994.

[58] S.X. Ju, M.J. Black, and Y. Yacoob, “Cardboard People: A Parametrized Model of Articulated Image Motion,” Second Int’l Conference. Automatic Face and Gesture Recognition, pp. 38-44, Oct. 1996.

[59] Y. Yacoob and M.J. Black, “Parametrized Modeling and Recognition of Activities,” Computer Vision and Image Understanding, vol. 73, no. 2, pp. 232- 247, 1999.

[60] R. Polana and R. Nelson, “Detecting Activities,” IEEE Conf. Computer Vision and Pattern Recognition, June 1993.

[61] R. O. Duda, Peter E. Hart y D. G. Stork. “Pattern Classi_cation, da. edici_on”. New York, Wiley-Interscience, Nov. 2001.

[62] M. Everingham and B. Thomas. S”upervised Segmentation and Tracking of Nonrigid Objects Using a “Mixture of Histograms" Model”. IEEE International conference on Image Processing, vol. 1 pp. 62-65, Oct. 2001.

Anexos

A. Manual del programador

Fución video_info:

```
void video_info(CvCapture* entrada, char* cad )

/*********************************************************************
Función: Muestra por pantalla la información concerniente al vídeo.

Parámetros de entrada:
CvCapture* entrada -Puntero a la captura del video.
char* cad -Nombre del vídeo.

Retorno:
*********************************************************************/
```

Fución matriz_frames:

```
void matriz_frames(CvCapture* entrada, CvMat** video, CvMat** video_R,
CvMat** video_G, CvMat** video_B, double num_frames, char *cad, int
flag )

/*********************************************************************
Función: Devuelve cuatro conjuntos de matrices que contiene los datos
de los frames del vídeo. El 1º conjunto contiene el vídeo en escala de
grises. El 2º conjunto contiene la componente primaria de color Rojo
de la secuencia de vídeo. El 3º conjunto contiene la componente
primaria de color azul de la secuencia de vídeo. El 4º conjunto
contiene la componente primaria de color verde de la secuencia de
vídeo.

Parámetros de entrada:
      CvCapture* entrada - puntero a la captura del video.
      double num_frames - número de frames del video que contiene la
      acción ejemplo.
char* cad - Nombre del vídeo.
      int flag - falg indicador de la etapa en la que se encuentra el
      algoritmo.

Retorno:
      CvMat** video - set de matrices que contiene el vídeo en escala
      de grises.
      CvMat** video_R - set de matrices que contiene la componente
      primaria de color Rojo de la secuencia de vídeo.
      CvMat** video_G - set de matrices que contiene la componente
      primaria de color Rojo de la secuencia de vídeo.
      CvMat** video_B - set de matrices que contiene la componente
      primaria de color Rojo de la secuencia de vídeo.
```

```
*****************************************************************/
```

Fución moving_window:

```
void moving_window(CvCapture* entrada, CvMat** video, CvMat** video_R,
CvMat** video_G,  CvMat** video_B, double num_frames, int flag)


/*****************************************************************
Función: Devuelve cuatro conjuntos de matrices que contiene los datos
de los frames del vídeo, después de descartar el primer frame del
cunjunto perteneciente a la iteración anterior y añadir un nuevo frame
al final del conjunto de matrices. El 1º conjunto contiene una
subsecuencia del vídeo en escala de grises. El 2º conjunto componente
primaria de color Rojo de una subsecuencia del vídeo. El 3º conjunto
componente primaria de color azul de una subsecuencia del vídeo. El 4º
conjunto componente primaria de color verde de una subsecuencia del
vídeo.

 Parámetros de entrada:
      CvCapture* entrada - puntero que apunta la captura del video.
      double num_frames - número de frames del video que contiene la
      acción ejemplo.
      int flag - falg indicador de la etapa en la que se encuentra el
      algoritmo.
Retorno:
      CvMat** video - set de matrices que contiene una subsecuencia
      del vídeo en escala de grises.
      CvMat** video_R - set de matrices que contiene la componente
      primaria de color Rojo una subsecuencia del vídeo.
      CvMat** video_G - set de matrices que contiene la componente
      primaria de color Rojo una subsecuencia del vídeo.
      CvMat** video_B - set de matrices que contiene la componente
      primaria de color Rojo una subsecuencia del vídeo.

*****************************************************************/
```

Fución media_y_desviacion_tipica_fondo:

```
void    media_y_desviacion_tipica_fondo(CvCapture*    entrada,    double
num_frames, CvMat* mediafondo,CvMat* desv_tipica_fonfo )


/*****************************************************************
Función: Devuelve una matriz que representa la media del fondo y otra
matriz que nos da la desviacion típica del fondo.

Parámetros de entrada:
      CvCapture* entrada - puntero que apunta la captura del video.
      double num_frames - número de frames con los que vamos a
      trabajar para conseguir la media y la desviación típica.

Retorno:
      CvMat* mediafondo - Matriz CvMat media del fondo.
      desv_tipica_fonfo - Matriz CvMat desviación típica del fondo.

*****************************************************************/
```

Fución video_pasobajo:

```
void video_pasobajo(CvMat** video, CvMat** video_pb, double
num_frames)

/*********************************************************************
Función: Muestrea la secuencia de vídeo y seguidamente pasa un filtro
paso baja obteniendo así el siguiente escalón de la pirámide temporal.

Parámetros de entrada:
      CvMat** video - set de matrices que contiene una subsecuencia
      del vídeo.
      double num_frames - número de frames del video que contiene la
      acción ejemplo.

Retorno:
      CvMat** video_pb - set de matrices que contiene el siguiente
      nivel de la pirámide temporal de una subsecuencia del vídeo.
*********************************************************************/
```

Fución gradiente_ejes_xyt:

```
void gradiente_ejes_xyt(CvMat** video_R, CvMat** video_G, CvMat**
video_B, double num_frames, IplImage** gradiente_t, CvMat** Nx,
CvMat** Ny, CvMat** Nt, int flag)

/*********************************************************************
Función: Calcula los gradiente en los ejes 'x, 'y' y tiempo para una
secuencia de vídeo, devolviendo el array de imágenes que forman el
gradiente temporal y los tres conjuntos de matrices que contienen los
gradientes espacio-temporales normalizados.


 Parámetros de entrada:
      CvMat** video_R - set de matrices que contiene la componente
      primaria de color Rojo una subsecuencia del vídeo.
      CvMat** video_G - set de matrices que contiene la componente
      primaria de color Rojo una subsecuencia del vídeo.
      CvMat** video_B - set de matrices que contiene la componente
      primaria de color Rojo una subsecuencia del vídeo.
      double num_frames - número de frames del video que contiene la
      acción ejemplo.
      int flag - falg indicador de la etapa en la que se encuentra el
      algoritmo.

Retorno:

      IplImage** gradiente_t - array de estructura IplImage que
      contiene la información del gradiente temporal.
      CvMat** Nx - set de matrices que contiene los datos de la
      normalización del gradiente espacial x.
      CvMat** Ny - set de matrices que contiene los datos de la
      normalización del gradiente espacial y.
      CvMat** Nt - set de matrices que contiene los datos de la
      normalización del gradiente temporal.

*********************************************************************/
```

Fución movimientos_independientes:

```
void movimientos_independientes(doublé num_frames, CvMat* mediafondo,
CvMat*      desv_tipica_fonfo,      CvMat**      video,      GRUPO_RECT
grupos_rectangulos[], IplImage** seg_BN )


/*********************************************************************
Función: Realiza una segmentación en blanco y negro para la detección
del frente de de los frames de un vídeo, a continuación independiza
los distintos movimientos que aparecen en la escena y devuelve un
array con los distintos grupos de rectángulos que representan cada una
de las acciones que suceden en la secuencia.

Parámetros de entrada:
      double num_frames - número de frames del video que contiene la
      acción ejemplo.
      CvMat* mediafondo - Matriz CvMat media del fondo.
      CvMat* desv_tipica_fonfo - Matriz CvMat desviación típica del
      fondo.
      CvMat** video - set de matrices que contiene una subsecuencia
      del vídeo en escala de grises.

Retorno:
      GRUPO_RECT grupos_rectangulos[] - array de estructura GRUPO_RECT
      que contiene de forma independiente los distintos movimientos
      que aparecen en la escena.
      IplImage** seg_BN - array de estructura IplImage que contiene la
      información de la segmentación para la detección del frente
      realizada sobre el vídeo.

*********************************************************************/
```

Fución histograma:

```
void histograma (IplImage** gradiente_t, CvMat** Nx, CvMat** Ny,
CvMat** Nt, double num_frames,int* hist_bins_x,int* hist_bins_y,int*
hist_bins_t, GRUPO_RECT *grupos_rectangulos, int flag, IplImage**
seg_BN)

/*********************************************************************
Función: Calcula los histogramas que contienen los datos que
caracterizan en movimiento.

Parámetros de entrada:
      IplImage** gradiente_t - array de estructura IplImage que
      contiene la información del gradiente temporal.
      CvMat** Nx - set de matrices que contiene los datos de la
      normalización del gradiente espacial x.
      CvMat** Ny - set de matrices que contiene los datos de la
      normalización del gradiente espacial y.
      CvMat** Nt - set de matrices que contiene los datos de la
      normalización del gradiente temporal.
      double num_frames - número de frames del video que contiene la
      acción ejemplo.
      int flag - falg indicador de la etapa en la que se encuentra el
      algoritmo.
      IplImage** seg_BN - array de estructura IplImage que contiene la
      información de la segmentación para la detección del frente
      realizada sobre el vídeo.
```

```
      GRUPO_RECT   *grupos_rectangulos   -   Cotien   un   conjunto   de
      rectángulos con los datos de un únco movimiento.


Retorno:
      int  hist_bins_x - array de int[256] que representa el
      histograma de los datos que se refieren al eje 'x'.
      int  hist_bins_y - array de int[256] que representa el
      histograma de los datos que se refieren al eje 'y'.
      int  hist_bins_t - array de int[256] que representa el
      histograma de los datos que se refieren al eje 't'.

**********************************************************************/
```

Fución distancia2:

```
double distancia2(int* hist_bins_x1_1,int* hist_bins_y1_1,int*
hist_bins_t1_1, int* hist_bins_x1_2,int* hist_bins_y1_2,int*
hist_bins_t1_2, int* hist_bins_x2_1,int* hist_bins_y2_1,int*
hist_bins_t2_1, int* hist_bins_x2_2,int* hist_bins_y2_2,int*
hist_bins_t2_2, int* hist_bins_x3_1,int* hist_bins_y3_1,int*
hist_bins_t3_1, int* hist_bins_x3_2,int* hist_bins_y3_2,int*
hist_bins_t3_2)


/**********************************************************************
Función Recibe 9 histogramas por cada acción, uno por cada una de las
dimensiones y escalas temporales y devuelve el valor de la distancia
de comportamiento entre las dos actividades analizadas

Parámetros de entrada:
      Conjunto e 9 histogramas de la acción ejemplo.
      Conjunto de 9 histogramas de la acción estudiada.
Retorno:
      double Distancia - distancia conductual entre dos movimientos.

**********************************************************************/
```

B. Conjunto de vídeos

Videos actividad ejemplo	
Nombre	Descripción
Caminar_ej1	Vídeo en el escenario correspondiente al experimento A, en el que aparece una persona realizando la acción caminar una vez con cada pierna.
Caminar_ej2	Vídeo en el escenario correspondiente al experimento B, en el que aparece una persona realizando la acción caminar una vez con cada pierna.
Caminar_ej3	Vídeo en el escenario independiente (experimento C), en el que aparece una persona realizando la acción caminar una vez con cada pierna.
Correr_ej1	Vídeo en el escenario correspondiente al experimento A, en el que aparece una persona realizando la acción correr una vez con cada pierna.
Correr_ej2	Vídeo en el escenario correspondiente al experimento B, en el que aparece una persona realizando la acción correr una vez con cada pierna.
Saltar_ej1	Vídeo en el escenario correspondiente al experimento B, en el que aparece una persona realizando la acción saltar.

Videos Analizados	
Nombre	Descripción
Caminar1_expA	Vídeo en el escenario correspondiente al experimento A, en el que aparece la persona_1 realizando la acción caminar de una lado a otro de la escena.
Caminar2_expA	Vídeo en el escenario correspondiente al experimento A, en el que aparece la persona_2 realizando la acción caminar de una lado a otro de la escena.
Correr1_expA	Vídeo en el escenario correspondiente al experimento A, en el que aparece la persona_1 realizando la acción correr de una lado a otro de la escena.
Correr2_expA	Vídeo en el escenario correspondiente al experimento A, en el que aparece la persona_2 realizando la acción correr de una lado a otro de la escena.
Lateral1_expA	Vídeo en el escenario correspondiente al experimento A, en el que aparece la persona_1 realizando la acción pasos laterales de una lado a otro de la escena.
Lateral2_expA	Vídeo en el escenario correspondiente al experimento A, en el que aparece la persona_2 realizando la acción pasos laterales de una lado a otro de la escena.
Caminar1_expB	Vídeo en el escenario correspondiente al experimento B, en el que aparece una persona realizando la acción caminar de una lado a otro de la escena.
Correr1_expB	Vídeo en el escenario correspondiente al experimento B, en el que aparece una persona realizando la acción correr de una lado a otro de la escena.
Lateral1_expB	Vídeo en el escenario correspondiente al experimento B, en el que aparece una persona realizando la acción pasos laterales de una lado a otro de la escena.
Gatear1_expB	Vídeo en el escenario correspondiente al experimento B, en el que aparece una persona realizando la acción gatear de una lado a otro de la escena.
Secuencia general_1	Vídeo en el escenario correspondiente al experimento B, en el que aparecen dos personas caminando y una apunta con un arma a la otra.
Secuencia general_2	Vídeo en el escenario correspondiente al experimento B, en el que aparecen dos personas caminando una detrás de otra.
Secuencia general_3	Vídeo en el escenario correspondiente al experimento B, en el que aparecen dos personas una realizando pasos laterales y otra persona simultáneamente está gateando.

Videos Analizados	
Nombre	Descripción
Secuencia general_4	Vídeo en el escenario correspondiente al experimento B, en el que aparecen de inicio dos personas una realizando la acción gatear y una de ellas cambia en la mitad del escenario la acción gatear por la acción caminar, mientras que la otra persona continua gateando.
Secuencia general_5	Vídeo en el escenario correspondiente al experimento B, en el que aparece una persona realizando la acción correr, en la mitad de la secuencia tropieza, se cae, se levanta y continua corriendo.
Secuencia general_5	Vídeo en el escenario correspondiente al experimento B, en el que aparece una persona alternando la acción correr y saltar.
Fondo multimodal_1	Vídeo en un escenario multimodal (playa), en el que aparece una persona realizando la acción caminar.
Fondo multimodal_2	Vídeo en un escenario multimodal (playa), en el que aparecen varias personas realizando la acción caminar.
Caminar Zoom_x1_1	Vídeo en el escenario correspondiente al experimento A, en el que aparece la persona_1 realizando la acción caminar de un lado a otro de la escena, modificando la distancia con respecto 5m.
Caminar Zoom x1 2	Vídeo en el escenario correspondiente al experimento A, en el que aparece la persona_2 realizando la acción caminar de un lado a otro de la escena, modificando la distancia con respecto 5m.
Caminar Zoom x1 1	Vídeo en el escenario correspondiente al experimento A, en el que aparece la persona_1 realizando la acción caminar de un lado a otro de la escena, modificando la distancia con respecto 10m.
Caminar Zoom_x1_2	Vídeo en el escenario correspondiente al experimento A, en el que aparece la persona_2 realizando la acción caminar de un lado a otro de la escena, modificando la distancia con respecto 10m.
Caminar 30º	Vídeo en el escenario correspondiente al experimento A, en el que aparece una persona caminando con un ángulo de 30º con respecto la cámara.
Caminar 60º	Vídeo en el escenario correspondiente al experimento A, en el que aparece una persona caminando con un ángulo de 60º con respecto la cámara.
Caminar 90º	Vídeo en el escenario correspondiente al experimento A, en el que aparece una persona caminando con un ángulo de 90º con respecto la cámara.

Presupuesto:

1) Ejecución Material

• Compra de ordenador personal (Software incluido) ..... 1.950 €

• Material de oficina..... 200 €

• Total de ejecución material ..... 2.150 €

2) Gastos generales

• 16 % sobre Ejecución Material ..... 344 €

3) Beneficio Industrial

• 6 % sobre Ejecución Material ..... 129 €

4) Honorarios Proyecto

• 1.200 horas a 15 € / hora ..... 18.000 €

5) Material fungible

• Gastos de impresión ..... 85 €

• Encuadernación ..... 200 €

6) Subtotal del presupuesto

• Subtotal Presupuesto ..... 20.908 €

7) I.V.A. aplicable

• 16% Subtotal Presupuesto ..... 3.345,28 €

8) Total presupuesto

• Total Presupuesto ..... **24.253,28 €**

Madrid, Febrero 2010
El Ingeniero Jefe de Proyecto

Fdo.: Pablo Manuel Herranz Fernández
Ingeniero Superior de Telecomunicación

PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un reconocedor de actividades en video basado en ejemplos. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se

consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad “Presupuesto de Ejecución de Contrata” y anteriormente llamado ”Presupuesto de Ejecución Material” que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.