Universidad Zaragoza
1542

Proyecto Fin de Carrera
Ingeniería Industrial

Segmentación temporal y reconocimiento débilmente supervisado de acciones en vídeos

Autor
Alberto Belled Casabona

Director
Javier Civera Sancho

Escuela de Ingeniería y Arquitectura
2014

A David, dondequiera que esté.
No sabes cuánto te echo de menos.

Agradecimientos

Hay muchas personas a las que me gustaría dar las gracias. A algunas, porque este proyecto no hubiera sido posible sin ellas. A otras, por motivos que van mucho más allá.

Gracias, en primer lugar, a mi director de proyecto, D. Javier Civera Sancho. Por su trabajo, su paciencia, su apoyo y su disponibilidad permanente para resolver cualquier duda o problema.

Gracias a mi madre. Por demostrarme lo valiente y fuerte que puede llegar a ser una persona. Por desear siempre lo mejor para mí. Por quererme más que a ella misma.

Gracias a mi padre. Por sus consejos y reprimendas. Por su confianza incluso en la adversidad. Por hacerme ver la clase de padre que quiero ser algún día.

Gracias a mis abuelos. Por sus historias y anécdotas. Por ese cariño difícil de explicar con palabras. Por quererme como se quiere a un hijo.

Gracias a mi tío, Manuel. Por enseñarme que, a pesar de los interminables baches que la vida va creando en el camino, es posible levantarse cada día con ilusión y una sonrisa en la cara. Por ser un verdadero superhéroe que no necesita capa ni poderes.

A mis amigos, a todos ellos. Por hacer más llevaderos los malos momentos e inolvidables los buenos. No hay necesidad de nombres, ellos saben bien quiénes son. Ojalá algún día pueda ser tan buen amigo como lo son ellos para mí.

Por último, gracias a Sara. Por el maravilloso tiempo a su lado. Por agarrarme bien fuerte en un momento que hubiera derrotado incluso al más fuerte de los mortales. Por enseñarme que los momentos más bonitos de nuestras vidas son aquéllos que se recuerdan con pocas palabras y muchas sonrisas.

Sinceramente,
Muchas gracias a todos

Segmentación temporal y reconocimiento débilmente supervisado de acciones en vídeos

RESUMEN

El reconocimiento de acciones en vídeos es, sin duda, uno de los problemas de visión por computador más relevantes en la actualidad. Uno de los principales motivos de que ésto sea así son las numerosas aplicaciones derivadas que podrían ser desarrolladas en diversos ámbitos de la ciencia y la vida cotidiana y el entretenimiento.

Si además de reconocer las acciones presentes en los vídeos somos capaces de segmentarlas temporalmente, ésto es, determinar los instantes en que empiezan y acaban, su identificación es mucho más completa. No sólo sabríamos que en el vídeo en cuestión aparece una determinada acción, sino que dispondríamos de información adicional para analizarla con más detalle.

En este proyecto se formula el problema de la segmentación temporal y el reconocimiento de acciones en vídeos mediante una función de coste, o función de energía, definida de forma débilmente supervisada. A diferencia de los métodos existentes, los cuales emplean un número enorme de vídeos anotados para entrenar los algoritmos, en este proyecto se ha utilizado un único vídeo anotado por cada acción que se pretende reconocer. Con ello conseguimos que la fase de aprendizaje del algoritmo sea menos costosa en esfuerzo humano y que el método sea aplicable a casi cualquier dataset de vídeos.

La energía formulada se compone de una serie de términos y parámetros que han sido ajustados mediante la experimentación. Se ha utilizado para ello un dataset de videos realistas extraídos de películas, construído a partir del dataset *Hollywood2*. La minimización de la energía proporciona la solución de menor coste del problema, es decir, la solución óptima. La bondad de los resultados de minimización se ha evaluado mediante la comparación con un *ground truth* creado a partir de los vídeos de estudio.

Los resultados obtenidos en nuestro dataset y en el dataset *KTH* demuestran que es posible obtener buenas tasas de acierto en segmentación temporal y reconocimiento de acciones en vídeos de forma débilmente supervisada.

Índice general

Capítulo 1

Introducción

1.1. Motivación

El objetivo último de la inteligencia artificial es la creación de máquinas inteligentes. Máquinas que tomen decisiones de la misma forma que las tomaría un ser humano, con la mayor recompensa posible. En otras palabras, se quiere conseguir que las máquinas «piensen».

Si una persona es la encargada de la toma de decisiones podrá aplicar su sentido común o sus conocimientos en la materia en cuestión para determinar la mejor vía de actuación. Sin embargo, la inteligencia artificial no está todavía lo suficientemente avanzada como para garantizar que las decisiones tomadas por un sistema automático, en base a cierta información obtenida de su entorno, sean siempre las idóneas.

La visión por computador es uno de los subcampos fundamentales de la inteligencia artificial. Incluye métodos para adquirir, procesar, analizar y comprender imágenes con el objetivo de extraer información a partir de ellas. Usando la misma analogía que en el primer párrafo, el objetivo es que las máquinas «vean».

Desde los inicios de la visión por computador el problema del reconocimiento en imágenes y vídeos ha sido uno de los más estudiados. El interés que el reconocimento genera se debe principalmente a la cantidad de potenciales aplicaciones. No es una exageración afirmar que la visión por computador es una de las disciplinas que mayor impulso y desarrollo ha experimentado en los últimos años. En la figura 1.1 se recogen algunas de las aplicaciones más recientes y punteras de la visión por computador y el reconocimiento.

Entre los problemas de reconocimiento visual, uno de los más estudiados es el reconocimiento de objetos. Las personas somos capaces de reconocer la

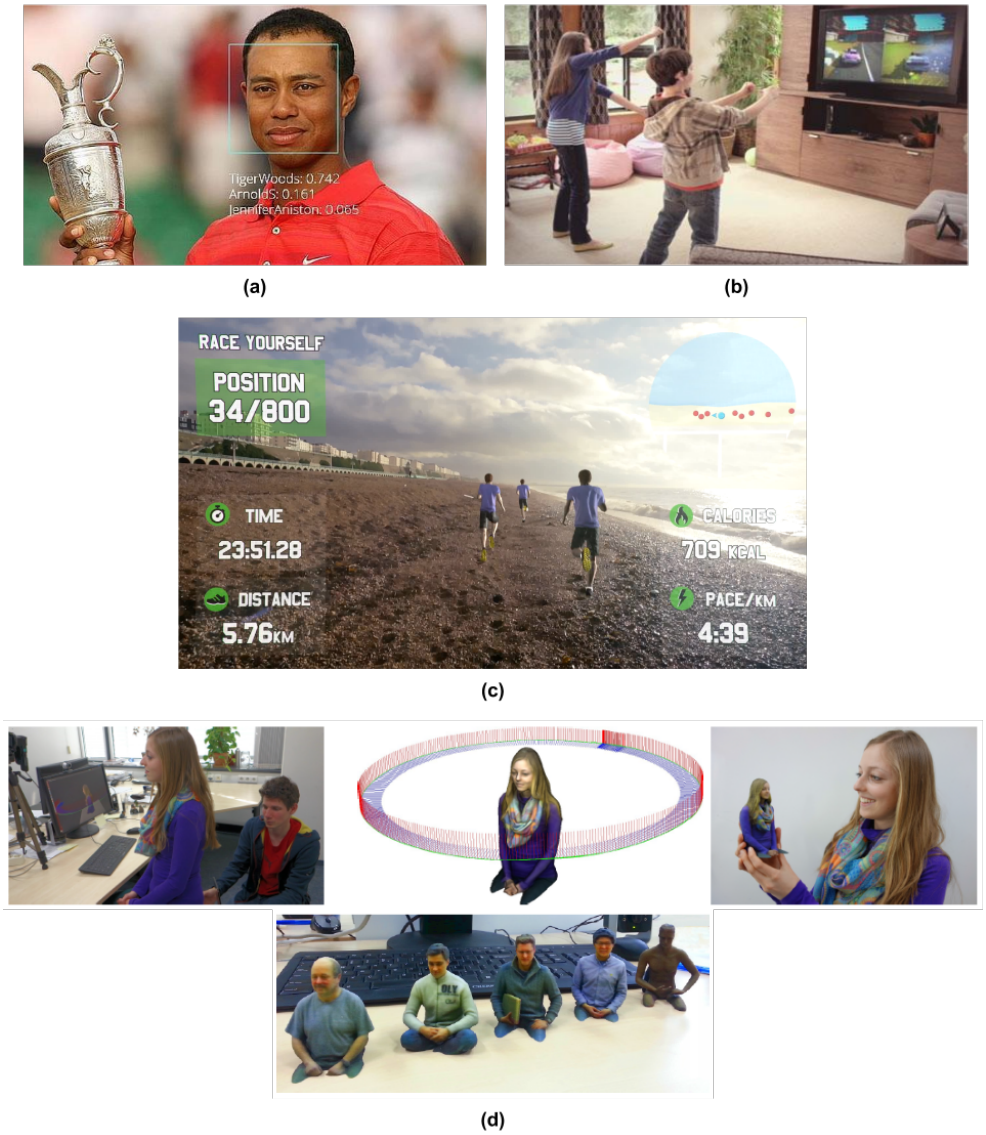

Figura 1.1: Ejemplos de aplicación de la visión por computador y el reconocimiento. (a) muestra un recocimiento de cara llevado a cabo por la aplicación diseñada por Google a tal efecto. En (b) aparecen dos niños jugando a la videoconsola sin necesidad de mandos, gracias al dispositivo *Kinect* de Microsoft. (c) presenta un posible uso de las *Google Glass*: la simulación de una carrera virtual. De esta forma los entrenamientos de los corredores serían más amenos. Por último, en (d) se muestra la aplicación *CopyMe3D* desarrollada por J. Sturm et al. [1]. Mediante un sensor RGB-D es posible realizar la reconstrucción 3D de una persona, la cual puede ser impresa mediante una impresora 3D.

mayoría de objetos en las imágenes sin apenas esfuerzo, a pesar de que dichos objetos pueden presentar variaciones en cuanto al punto de vista, a la escala e incluso pueden encontrarse desplazados o rotados. Somos capaces de reconocerlos incluso cuando una parte de ellos está oculta a la vista. Sin embargo, estas tareas tan sencillas para nosotros suponen todavía un reto para los sistemas de visión por computador.

Hoy en día las cámaras digitales son sensores con un coste muy reducido que además pueden ser instalados en muchos sistemas con la finalidad de extraer información del entorno que las rodea, de forma continua. La tecnología digital está de hecho tan desarrollada que las imágenes o vídeos capturados actualmente por una cámara pueden llegar a tener una calidad muy elevada. Las nuevas pantallas y dispositivos con resolución 4k son un buen ejemplo de ello.

La información obtenida será procesada e interpretada a posteriori y un sistema automático o humano se encargará de tomar decisiones en base a los resultados obtenidos. Es por tanto fundamental que el procesamiento y la interpretación de la información visual sean lo más avanzados posible, para facilitar así la toma de decisiones del sistema encargado, ya sea una persona o un autómata.

Los algoritmos de reconocimiento de objetos actuales son capaces de conseguir unas tasas de acierto bastante altas, pero esta tendencia cambia cuando lo que se quiere reconocer son acciones llevadas a cabo por personas. El problema del reconocimiento de acciones supone un desafío mucho mayor para la visión por computador. Si bien se están consiguiendo tasas de acierto altísimas en el reconocimiento de acciones en imágenes estáticas y vídeos sencillos, los resultados en el reconocimiento de acciones en vídeos más complejos y realistas distan mucho de ser los requeridos para hacer que una máquina entienda perfectamente las acciones que están teniendo lugar frente a ella.

Aunque complicado y todavía en los primeros estadios de la investigación, el reconocimiento de acciones en videos es un problema con una gran cantidad de potenciales aplicaciones en diversos ámbitos tanto de la ciencia como de la vida cotidiana.

Pensemos por un momento en las personas invidentes. Existen ya sistemas de visión por computador capaces de analizar el entorno que las rodea con el objetivo de proporcionar información detallada de posibles obstáculos o vías de acceso. Pero, ¿qué pasaría si esa persona quisiera tener una relación más estrecha con el mundo que la rodea y ser capaz de conocer las acciones que están realizando las personas a su alrededor? A la hora de ver una película, ¿no sería interesante que ésta no estuviera adaptada y fuera el sistema de ayuda a la visión el que se encargara de mantener a la persona al corriente de las

acciones que están apareciendo?.

Con el paso de los años los robots de asistencia serán comunes en viviendas y hospitales. ¿Quién no querría que, para una mejor interacción con los usuarios, fueran capaces de entender de forma precisa sus necesidades? A lo mejor el enfermo tiene intención de incorporarse. Quizá la persona es minúsvalida y no es capaz de coger el teléfono que está sonando. Puede ser, poniéndonos en el peor caso, que la persona se sienta mal y necesite llamar a una enfermera, pero no consiga llegar al hasta el botón de llamada. En todas estas situaciones sería de gran ayuda que el robot fuera capaz de comprender las acciones que la persona a su cargo está llevando a cabo, o intentándolo.

Para evaluar los métodos de reconocimiento de acciones se suelen utilizar datasets estándar, algunos de los cuales tienen miles de vídeos, clasificados en decenas de categorías o acciones diferentes. Estos datasets suelen estar anotados, es decir, se proporcionan archivos que recogen la acción contenida en cada vídeo, lo que facilita el ajuste de los algoritmos creados. Las dificultades surgen cuando se quieren diseñar datasets propios o extraer los vídeos de la red: anotar estos vídeos suele ser una tarea tediosa y cara, tanto en tiempo como en recursos. Imaginemos que una persona experimentada es capaz de anotar de forma precisa 50 vídeos en una hora. Si se desea construir un dataset de decenas de miles de vídeos extraídos, por ejmplo, de *Youtube*, con el objetivo de probar el funcionamiento de un algoritmo propio o para que otros investigadores puedan evaluar los suyos, los costes se disparan.

Es por este motivo que se han desarrollado algoritmos de aprendizaje débilmente supervisado, es decir, con la menor carga de trabajo posible para las personas. Estos algoritmos de reconocimiento, para el caso concreto del reconocimiento de acciones, en lugar de anotar cientos de vídeos, son capaces de obtener resultados satisfactorios utilizando sólo unos pocos vídeos anotados y una gran cantidad de vídeos sin anotar. Aquí radica la principal ventaja de nuestro método y la característica que busca diferenciarlo del resto de algoritmos existentes: se va a utilizar únicamente un vídeo anotado por acción.

Además del concepto de supervisión débil nuestro trabajo introduce también el idea de la segmentación de acciones, entendida desde un punto de vista temporal. Si además de reconocer la acción que está teniendo lugar en un vídeo somos capaces de acotarla en el dominio de sus frames, determinando cuándo empieza y cuándo acaba, el reconocimiento será mucho más completo. El presente proyecto presenta un método para llevar a cabo identificaciones de acciones en vídeos realistas (extraídos de películas) mediante un proceso de minimización que combina segmentación y reconocimiento.

1.2. Estado del arte

El interés que el reconocimiento de las acciones llevadas a cabo por personas en vídeos ha despertado en los últimos años en el ámbito de la visión por computador está motivado por múltiples aplicaciones, tanto «online» como «offline», y la existencia de grandes datasets de vídeos como *Youtube* [2]. El reconocimiento automático de las acciones en los vídeos posibilita búsquedas mucho más eficientes de, por ejemplo, derribos en los partidos de fútbol, apretones de manos en los montajes de noticias de los telediarios o movimientos típicos de baile en vídeos musicales. El procesamiento online permite, yendo un poco más lejos, la realización de estudios automáticos en edificios y establecimientos e incluso desarrollar funciones de ayuda para gente anciana en viviendas domóticas.

La principales dificultades que el reconocimiento de acciones ha presentado desde sus inicios radican en las enormes variaciones individuales de las personas en cuanto a su expresión, postura, movimiento y vestuario; efectos derivados de la perspectiva y el movimiento relativo de la cámara; variaciones de iluminación y efectos del fondo.

Por ello, a lo largo de los años se han realizado numerosas suposiciones y se han impuesto restricciones, como pueden ser movimientos acotados de la cámara, contextos de escena específicos, variaciones restringidas de los puntos de vista y, muy frecuentemente, independencia de la acción con el fondo de la escena, algo que no siempre es cierto.

Los primeros métodos desarrollados se basaban en el seguimiento de ciertas partes del cuerpo al efectuar los movimientos y en la evolución de las siluetas de los individuos en el tiempo. Estos métodos asumían que las acciones podían ser reconocidas a partir de los contornos de los cuerpos, lo que no siempre es posible.

En torno al año 2000 comenzaron a hacerse populares los métodos de reconocimiento de acciones basados en medidas locales en puntos de las imágenes con interés espacio-temporal. Estos descriptores, como se demuestra en [3], pueden capturar eventos locales en los vídeos y son adaptables al tamaño, velocidad y frecuencia de patrones en movimiento, lo que posibilita el reconocimiento de dichos patrones (acciones). Algunos métodos, como los descritos en [3] y [4], integran los descriptores locales en algoritmos basados en *Support Vector Machines* (SVM), los cuales tratan de encontrar el hiperplano que separa dos clases (el vector de descriptores y las etiquetas correspondientes). En [4] se introduce además la idea de estudiar la influencia del contexto de la escena en el reconocimiento de las acciones.

Existen también métodos de reconocimiento de acciones en vídeos que han sido pensados para identificar acciones concretas, como por ejemplo la acción

«beber» [5]. Estos métodos, que por primera vez se adentran en el reconocimiento en escenarios sin restricciones como son los de las películas de cine, están basados frecuentemente en combinaciones de descriptores espacio-temporales y clasificadores de *Keyframes*, y han demostrado ser muy eficaces a la hora de reconocer aquellas acciones para las que han sido implementados.

Evidentemente, al igual que [5] aplica su método a la acción «beber», éstos métodos podrían aplicarse al reconocimiento de cualquier otra acción tras realizar los ajustes pertinentes. Sin embargo, ésto supondría adaptar el método a cada una de las acciones conocidas que se quieran detectar, lo que implicaría llevar a cabo un exhaustivo proceso de entrenamiento y comprobación de los clasificadores. En muchas ocasiones el coste de estos procesos se minimiza mediante la creación de una bolsa de descriptores o palabras. Ésto permite aplicar métodos más complejos, como son los basados en patrones de cambio de movimiento, de forma más sencilla [6].

Técnicas más actuales de reconocimiento de acciones utilizan *Actons* [7], procesos de co-segmentación en pares de vídeos [8] o la información de profundidad obtenida con los cada vez más populares y extendidos sensores RGB-D [9]. A partir del mapa de profundidad se extraen descriptores que describen la posición 3D de las articulaciones de los individuos y pueden ser utilizados para extraer un diccionario de términos 3D. Sin embargo, a pesar de su popularidad, el uso exitoso de estos sensores está actualmente limitado al entretenimiento y a las interacciones humano-máquina. Además, estos sistemas no pueden ser utilizados para anotar vídeos RGB convencionales o identificar el movimiento desde una gran distancia, ya que tienen limitación de rango.

En lo referente a la segmentación temporal de las acciones no hay ningún trabajo relevante en la bibliografía estudiada. Se debe, en parte, a que los vídeos de la mayoría de los datasets utilizados contienen únicamente personas llevando a cabo las acciones y por lo tanto no hay necesidad de determinar el instante de inicio y final de las mismas. De igual forma, tampoco se han encontrado métodos que lleven a cabo un reconocimiento débilmente supervisado. Todos los estudiados utilizan un número elevado de vídeos anotados. Es por ello que el presente trabajo aborda el problema de la segmentación temporal y el reconocimiento débilmente supervisado de acciones en vídeos.

1.3. Objetivos y alcance

Se detallan a continuación los principales objetivos del presente proyecto:

- Desarrollar una ecuación de energía que permita atacar el problema del reconocimiento y la segmentación de acciones en vídeos. Dicha energía

se basa principalmente en descriptores espacio-temporales. Además, la energía se desarrollará para reconocer y segmentar, de forma débilmente supervisada, cualquier acción que se le presente.

- Crear un dataset de acciones para evaluar el método desarrollado. Para ello, los vídeos que lo compongan deberán presentar una gran variedad de acciones, escenarios, puntos de vista de la escena, iluminación y vestuario. Se quiere estar en disposición de reconocer, por ejemplo, un apretón de manos que tenga lugar en un restaurante con cientos de comensales y uno que ocurra en un discreto despacho donde los únicos individuos presentes sean los que llevan a cabo la acción.

- Obtener resultados que permitan demostrar que la segmentación temporal y el reconocimiento débilmente supervisado de acciones en vídeos realistas son posibles. Para que dichos resultados sean válidos, deberán haber sido obtenidos en datasets existentes o de gran complejidad.

- Definir, en base a los resultados y al estudio realizado, posibles líneas futuras de trabajo en el ámbito del reconocimiento y la segmentación de acciones en vídeos.

1.4. Estructura de la memoria

El proyecto se dividide en capítulos con el siguiente contenido:

- El capítulo 2 está dedicado a la formulación teórica del problema. Se describe de forma detallada la ecuación de energía creada para llevar a cabo el reconocimiento y la segmentación temporal de las acciones. Tras presentar cada uno de los términos que la componen se explica el método de minimización utilizado.

- El capítulo 3 presenta los resultados obtenidos en los experimentos llevados a cabo en el proyecto. Se describen detalladamente los ensayos realizados para determinar el valor óptimo de los parámetros de la ecuación de energía. El capítulo termina con un análisis de los resultados obtenidos al evaluar la energía definida y ya parametrizada, tanto en nuestro dataset como en otro famoso dataset para el reconocimiento de acciones en vídeos.

- Finalmente, el capítulo 4 presenta una síntesis del trabajo realizado y propone líneas futuras de investigación.

La memoria incluye también varios anexos:

- El anexo A recoge las anotaciones de los vídeos que componen nuestro dataset.
- En el anexo B se documenta el código desarrollado en el presente trabajo. Los archivos y directorios más relevantes son brevemente descritos con la finalidad de que el lector pueda hacerse una idea del funcionamiento y los pasos a seguir a la hora de ejecutar nuestro algoritmo.

Capítulo 2

Formulación del problema

2.1. Introducción. *Energy Functional*

Muchos problemas en visión por computador pueden ser formulados como la minimización de un apropiado *energy functional*, también denominado *cost functional*. La formulación continua de un *energy functional* es de la forma:

$$E(u) = \int_{\Omega} L(u, \nabla u, x)dx \quad \text{donde} \quad x \in \Omega, \quad u : \Omega \to \Re \tag{2.1}$$

En el cálculo variacional, un *functional* es una función o aplicación de un espacio vectorial en un campo escalar, o conjunto de funciones de los números reales. En otras palabras, es una función que toma un vector como argumento de entrada y devuelve un escalar.

Comúnmente, el espacio vectorial es un espacio de funciones. Por lo tanto, el *functional* toma una función como argumento y ésto hace que sea considerado una función de funciones. Buscamos, en este caso, la solución continua o función, *u*, que minimiza el *functional*, *E(u)*. En lo sucesivo nos referiremos al *energy functional* simplemente como «ecuación de energía», «función de energía», «función de coste» o «energía» a secas.

Existen muchos problemas en visión por computador resueltos mediante la minimización de una energía. Algunos ejemplos interesantes son: la segmentación de imágenes [10], la reconstrucción estéreo a grandes escalas (calles, monumentos, ciudades...) [11] o la reconstrucción 3D a partir de una sola imagen [12]. En la figura 2.1 se pueden ver ejemplos de estas aplicaciones.

La segmentación y el reconocimiento de acciones en vídeos no son una excepción y en el presente trabajo se presenta una resolución de este problema basada en la minimización de una ecuación de energía.

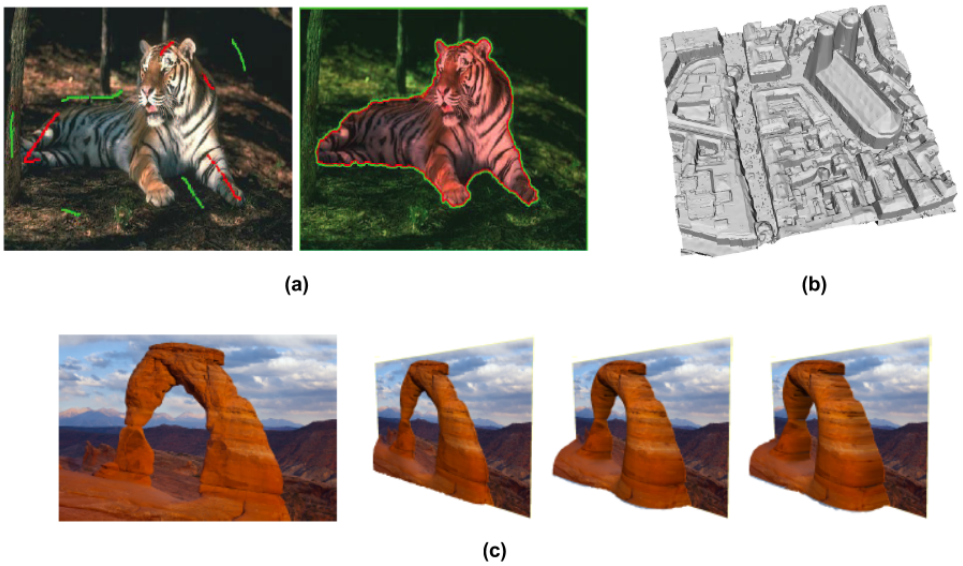

Figura 2.1: Ejemplos de aplicaciones de visión por computador que se basan en la minimización de una energía. (a) muestra la segmentación del tigre en la imagen original. En (b) se observa la reconstrucción de una ciudad llevada a cabo mediante un sistema estéreo. (c) demuestra que es posible «reconstruir» objetos en 3D a partir de una sola imagen de los mismos.

2.2. Ecuación de energía

Como ya se ha explicado, el principal objetivo del este trabajo es la segmentación temporal y el reconocimiento de acciones en vídeos. Dicho con otras palabras, se debe clasificar cada frame en alguna de las categorías, o acciones, contempladas (sentarse, comer, correr, no acción...) Ésto nos lleva a definir una función de coste de la forma:

$$E(L|\alpha) = \Psi + \Phi \tag{2.2}$$

donde

- Ψ es el término binario. Se trata del término que asigna a cada pareja de frames el coste de pertenecer a una categoría o categorías.
- Φ es el término unario. Determina, para cada frame del problema, el coste asociado a que dicho frame pertenezca a una de las categorías o acciones estudiadas.

De acuerdo con la formulación comúnmente utilizada en el cálculo variacional aplicado a la visión por computador, se puede denominar el término unario como *Data Term* y el binario como *Regularizer*.

El término binario, Ψ, se compone a su vez de dos términos asociados a descriptores de bolsa de palabras, uno basado en SIFT 2D, Ψ^{2D}, y otro basado en SIFT 3D, Ψ^{3D}:

$$\Psi = \lambda_1 \Psi^{2D} + \lambda_2 \Psi^{3D} \quad \text{con} \quad \lambda_2 = 1 - \lambda_1 \tag{2.3}$$

El término unario, por su parte, está formado por la suma de tres componentes:

$$\Phi = \Phi_1 + \Phi_2 + \Phi_3 \tag{2.4}$$

donde

- Φ_1 es el término unario derivado de la extracción de descriptores semánticos de bolsa de palabras. Se obtiene por la suma de dos componentes, una asociada a descriptores SIFT 2D, Φ^{2D}, y otra asociada a descriptores SIFT 3D, Ψ^{3D}:

$$\Phi_1 = \xi_1 \Phi_1^{2D} + \xi_2 \Phi_1^{3D} \quad \text{con} \quad \xi_2 = 1 - \xi_1 \tag{2.5}$$

- Φ_2 es el término unario derivado de la detección de personas en las imágenes.

- Φ_3 es el término unario de acciones centradas.

La ecuación de energía está definida sobre dos dominios:

- L. Conjunto de posibles etiquetas de los frames del problema. Para un dataset formado por N vídeos, siendo M_i el número de frames del vídeo i, el etiquetado puede expresarse de la forma:

$$L = (l_1^1, l_2^1, \ldots, l_{M_1}^1, \ldots, l_{M_N}^N) \tag{2.6}$$

- α. Conjunto de parámetros que gobiernan el peso relativo entre los términos que conforman la energía y sus componentes. Se determinan mediante entrenamiento.

$$\alpha = \{\lambda_1, \lambda_2, \xi_1, \xi_2, \delta_1, \delta_2, K_-, K_+, \epsilon_1, \epsilon_2\} \tag{2.7}$$

El resto del capítulo está dedicado al proceso de creación de los términos de la energía aquí presentados. Dicho proceso es inevitablemente inherente a la experimentación. Por ello, el lector debe entender que, aunque en el presente capítulo se den a conocer los términos de la energía, debe esperar al capítulo 3, en el que se desarrolla toda la experimentación llevada a cabo, para tener una visión global del proceso de definición de la ecuación de energía.

2.3. Términos basados en descriptores semánticos

Una de las tareas más importantes en visión por computador es la localización, en una imagen dada, de los puntos tanto relevantes en cuanto a la cantidad de información de su entorno como parcialmente estables frente a las perturbaciones y transformaciones locales y globales que puede sufrir la imagen. Esto último garantiza que dichos puntos se puedan detectar de forma repetible y precisa.

Uno de los algoritmos más utilizados para ello es el *Scale-Invariant Feature Transform*, SIFT [13], capaz de llevar a cabo una detección de características parcialmente invariantes tanto a rotaciones como al escalado de las imágenes.

Los elementos más importantes de nuestra ecuación de energía son aquellos que se basan en descriptores que nos proporcionan información de alto nivel a partir de información de muy bajo nivel, como los obtenidos mediante la aplicación del algoritmo SIFT. Esos descriptores se han denominado descriptores semánticos en el presente trabajo.

La finalidad principal de los descriptores semánticos es determinar el grado de similitud existente entre dos frames cualesquiera de nuestro dataset. Es lógico pensar que, si dos frames resultan ser muy parecidos en lo que a descriptores semánticos se refiere, es muy probable que contengan la misma acción o, por defecto, dos acciones similares.

Son estos descriptores los que permiten que nuestra ecuación de energía opere sin conocer las acciones llevadas a cabo en los vídeos, puesto que se limita, de alguna forma, a clasificar en diferentes categorías aquellos conjuntos de frames que han resultado ser similares semánticamente. La persona encargada de analizar los resultados conoce la acción asocida a cada una de las categorías, pero el algoritmo es «ciego» a esta información.

Se han implementado dos tipos de descriptores semánticos obtenidos a partir de la detección de características SIFT: descriptores espaciales, basados en SIFT 2D y descriptores espacio-temporales, basados en SIFT 3D. En las siguientes secciones se presentan estos descriptores, se explica el proceso de extracción asociado y se definen los términos de la energía derivados de su obtención.

2.3.1. Descriptores espaciales. SIFT 2D

Por defecto, el algoritmo SIFT trata de localizar los puntos de interés en una imagen, que es una matriz bidimensional. Por ello, nos podemos referir a

él como SIFT 2D. Si este algoritmo es aplicado en un mallado denso de localizaciones de una imagen, con una escala y orientación fijas, obtenemos los que se pueden denominar descriptores SIFT densos, muy utilizados en problemas como la categorización y el reconocimiento de objetos.

En nuestro trabajo se han implementado los descriptores PHOW [14, 15], que son una variación de los descriptores SIFT densos, extraídos a varias resoluciones (se han utilizado resoluciones 2, 4, 8 y 10). Como ya se ha indicado, el objetivo de extraer este tipo de descriptores en los frames de los vídeos del dataset es la obtención de información de alto nivel, semántica. La mejor forma de conseguirlo es la creación de una *Bag of Words*, BoW. Los términos o palabras que conformen esta bolsa servirán para definir los descriptores semánticos.

Extracción de los descriptores y BoW

El primer paso para crear una bolsa de palabras es el entrenamiento del vocabulario o conjunto de palabras visuales que la forman. Para llevar a cabo dicho entrenamiento se han extraído 540 frames de entre 180 vídeos del dataset Hollywood2 [16], 3 por vídeo. Los frames elegidos son el central, por ser muy probable que contenga la acción buscada y los frames centrales de las dos regiones en que queda dividido el vídeo, ya que es posible que contengan información acerca del principio y final de dicha acción (ver figura 2.2).

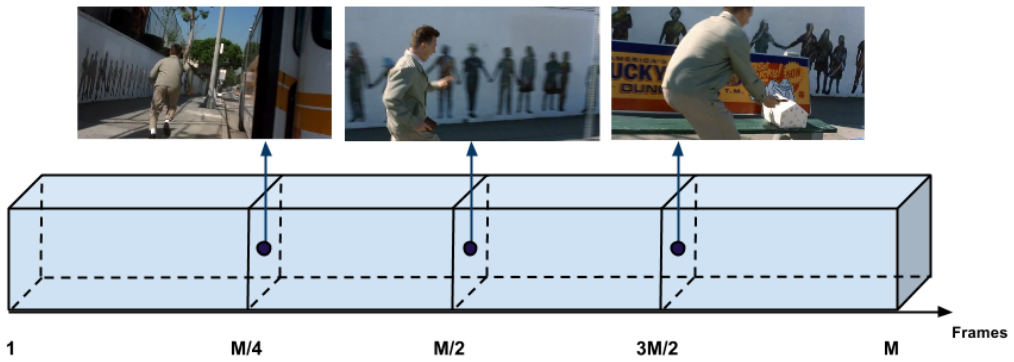

Figura 2.2: Frames seleccionados en cada vídeo para llevar a cabo el entrenamiento del vocabulario. Los tres frames, especialmente el central, son susceptibles de contener información acerca de la acción de estudio en ese vídeo.

Tras extraer los descriptores PHOW en los frames de entrenamiento, dichos descriptores deben ser cuantificados para obtener las palabras visuales. Esta cuantificación se ha llevado a cabo mediante la aplicación del algoritmo *K-means*. Se trata de un algoritmo de *clustering* que se encarga de dividir un conjunto de vectores en K grupos centrados en torno a un vector medio común

(figura 2.3). De esta forma, el proceso de *clustering* determina el vocabulario óptimo para cuantificar vectorialmente los datos, o descriptores PHOW en nuestro caso.

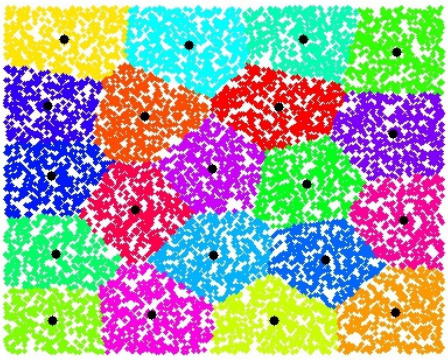

Figura 2.3: Ejemplo de un *clustering* de datos. Éstos se agrupan en torno a un dato medio central, o palabra.

El último paso del entrenamiento es la creación de un *kd-tree*, que consiste en una estructura de datos de particionado del espacio que organiza los puntos en un espacio euclídeo de k dimensiones. Esta estructura permitirá una computación mucho más rápida de los descriptores de bolsa de palabras en cada frame.

Una vez el vocabulario ha sido construído, se debe efectuar la extracción de los descriptores PHOW en todos los frames que conforman el dataset y calcular los histogramas espaciales asociados. Dichos histogramas, a los que a menudo nos referiremos como histogramas de descriptores, no son otra cosa que las frecuencias relativas con que las palabras del vocabulario aparecen en los frames. Para calcularlos se han considerado dos escalas: 2x2 y 4x4. La figura 2.4 ejemplifica esta consideración.

Figura 2.4: Escalas utilizadas en la extracción de los descriptores de bolsa de palabras en todos los frames. Al ser extraídos en 20 regiones diferentes de la imagen la información de alto nivel obtenida es mucho más rica.

El vocabulario construído consta de 600 palabras y los histogramas de descriptores se han obtenido en 20 regiones diferentes de las imágenes, por lo que el histograma resultante, para cada frame, s, es de la forma:

$$h(s) = \Big[h_1, ..., h_i, ..., h_{20}\Big], \quad h_i(s) = \Big[h_i^{w1}(s), \ldots, h_i^{w600}(s)\Big] \tag{2.8}$$

Estos histogramas serán utilizados para definir el término binario y el término unario asociados a los descriptores de bolsa de palabras, o semánticos, basados en SIFT 2D, como se verá a continuación.

Término binario asociado

Como se explicaba en 2.2, el término binario representa los costes asociados al hecho de que dos frames, s y q, pertenezcan a una determinada categoría o categorías. Puede ser entendido por tanto como el «coste de unión» de dichos frames.

La expresión propuesta para el término es la siguiente:

$$\Psi^{2D} = \sum_{s,q} \Delta_{s,q}^{BoW^{2D}}(l_s, l_q) \tag{2.9}$$

es decir, la distancia entre los descriptores de BoW de los frames s y q.

La ecuación 2.9 formula una comparación masiva entre los frames de todos los vídeos que componen nuestro dataset. Ésto es análogo a entender el término binario como una comparativa entre todos los vídeos de estudio, con el objetivo de encontrar aquellas regiones de frames con descriptores similares, pues esos frames serán los que, con una mayor probabilidad, contengan acciones iguales o similares. En la figura 2.5 se puede ver un esquema de estas comparaciones.

Al ser las comparaciones masivas, resulta lo mismo comparar un vídeo, i, con otro vídeo, j, que hacerlo al revés; la matriz es, por tanto, simétrica:

$$\Delta_{i,j}^{BoW^{2D}} = \Delta_{j,i}^{BoW^{2D}} \tag{2.10}$$

Además, los términos de la diagonal son nulos:

$$\Delta_{i,j}^{BoW^{2D}} = 0 \Leftrightarrow i = j \tag{2.11}$$

Es decir, no se contemplan las comparaciones entre frames de un mismo vídeo.

Con todas estas consideraciones, para un dataset de N vídeos, se obtiene un término binario asociado a los descriptores de bolsa de palabras de la forma:

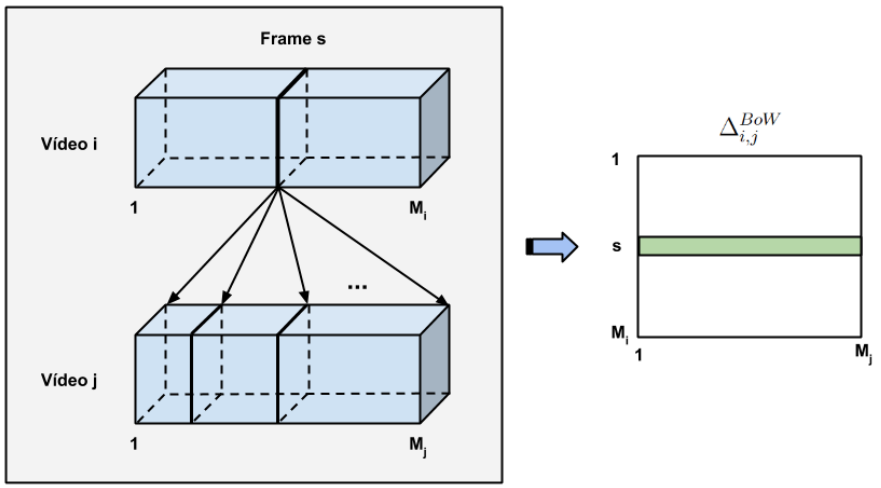

Figura 2.5: Esquema de las comparaciones de histogramas entre todos los frames del problema con la finalidad de definir el término binario de la energía asociado a los descriptores semánticos. La comparación del frame s del vídeo i con todos los frames del vídeo j define una fila de la submatriz asociada a dichos vídeos.

$$\Psi = \begin{bmatrix} 0 & \Delta_{1,2}^{BoW^{2D}} & \Delta_{1,3}^{BoW^{2D}} & \cdots & \Delta_{1,N}^{BoW^{2D}} \\ \Delta_{2,1}^{BoW^{2D}} & 0 & \Delta_{2,3}^{BoW^{2D}} & \cdots & \Delta_{2,N}^{BoW^{2D}} \\ \Delta_{3,1}^{BoW^{2D}} & \Delta_{3,2}^{BoW^{2D}} & 0 & \cdots & \Delta_{3,N}^{BoW^{2D}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Delta_{N,1}^{BoW^{2D}} & \Delta_{N,2}^{BoW^{2D}} & \Delta_{N,3}^{BoW^{2D}} & \cdots & 0 \end{bmatrix} \tag{2.12}$$

Término unario asociado

La función del término unario es evaluar la probabilidad de que un frame cualquiera, s, pertenezca a cada una de las categorías propuestas, C. En nuestro caso son la categoría *None*, que engloba todos aquellos frames que no contienen acción y las 12 categorías, o acciones, del dataset.

Su formulación es la siguiente:

$$\Phi_1^{2D} = \sum_s \Delta_s^{BoW^{2D}}(l_s, \Omega_i) \quad \forall i \in \Big[1, C\Big] \tag{2.13}$$

El método que se ha utilizado para evaluar dicha probabilidad es la comparación normalizada (ver figura 2.6) entre todos los frames de cada vídeo y los frames normalizados asociados de unos vídeos que han sido previamente

anotados, Ω_i (uno por cada acción a reconocer). Dado un frame cualquiera, s, de un video, i, su frame normalizado asociado, q, en el otro vídeo, j, se obtiene aplicando:

$$q = s^* = s\frac{M_i}{M_j} \tag{2.14}$$

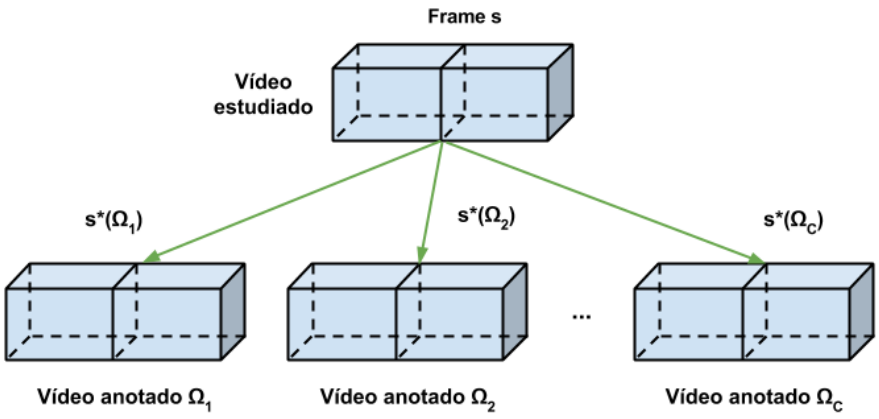

Figura 2.6: Esquema de las comparaciones con los vídeos anotados para construir el término unario de bolsa de palabras basado en SIFT 2D. Cada frame, s, del vídeo estudiado se compara con el frame normalizado asociado, $s*$, de cada uno de los vídeos anotados, Ω_i

.

De esta forma, el resultado de la comparación será el potencial asociado a cada frame; cuanto menor sea el resultado menor será este potencial y, por consiguiente, habrá una mayor probabilidad de que el frame pertenezca a la categoría del frame anotado. La matriz obtenida, para un número total de frames M, es de la forma:

$$\Phi_1^{2D} = \begin{bmatrix} \Delta_1^{BoW^{2D}}(l_1,\Omega_1) & \cdots & \Delta_s^{BoW^{2D}}(l_s,\Omega_1) & \cdots & \Delta_M^{BoW^{2D}}(l_M,\Omega_1) \\ \Delta_1^{BoW^{2D}}(l_1,\Omega_2) & \cdots & \Delta_s^{BoW^{2D}}(l_s,\Omega_2) & \cdots & \Delta_M^{BoW^{2D}}(l_M,\Omega_2) \\ \Delta_1^{BoW^{2D}}(l_1,\Omega_3) & \cdots & \Delta_s^{BoW^{2D}}(l_s,\Omega_3) & \cdots & \Delta_M^{BoW^{2D}}(l_M,\Omega_3) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Delta_1^{BoW^{2D}}(l_1,\Omega_C) & \cdots & \Delta_s^{BoW^{2D}}(l_s,\Omega_C) & \cdots & \Delta_M^{BoW^{2D}}(l_M,\Omega_C) \end{bmatrix} \tag{2.15}$$

2.3.2. Descriptores espacio-temporales. SIFT 3D

Las extensiones de los descriptores SIFT a un espacio de 2+1 dimensiones, siendo la tercera dimensión el tiempo, han sido estudiadas en el contexto del reconocimiento de acciones en vídeos [17]. Con este método los descriptores son extraídos en localizaciones de interés espacio-temporal, por lo que son más interesantes a la hora de describir acciones que tienen lugar en el tiempo. En la figura 2.7 se aclara este concepto y se presenta la principal diferencia entre los descriptores SIFT 2D y SIFT 3D.

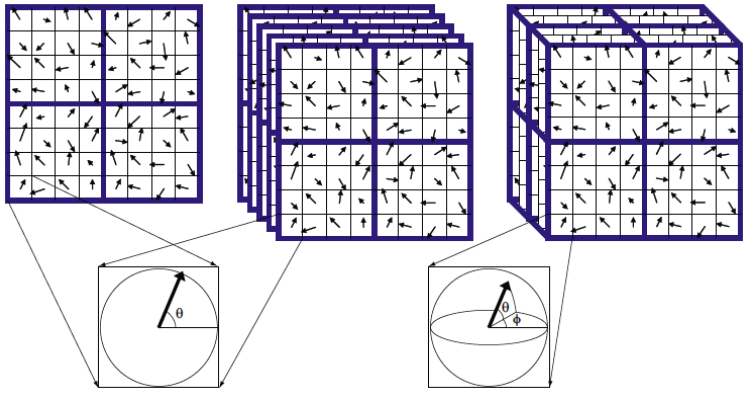

Figura 2.7: En la imagen izquierda se pueden ver descriptores SIFT 2D estándar. La imagen central muestra cómo estos descriptores pueden ser extraídos en todos los frames de un vídeo con la finalidad de describirlo espacialmente. Si consideramos la adición de una tercera dimensión y los gradientes asociados a los descriptores se calculan teniendo también en cuenta los frames vecinos obtenemos los descriptores SIFT 3D, como escenifica la imagen derecha. Ahora se tiene una «esfera de gradientes», frente a la «circunferencia de gradientes» de los descriptores SIFT 2D.

En lo que a la implementación de estos descriptores se refiere, la principal diferencia con los SIFT 2D radica en su computación. Por ello, se explicará de forma detallada el proceso de extracción llevado a cabo. En cuanto a los términos de la energía asociados, su construcción es idéntica y por ello referimos al lector a la sección 2.3.1 para conocer los métodos utilizados. Sin embargo, en aras de una mayor claridad y fácil comprensión, se presentará la notación asociada.

Extracción de los descriptores y BoW

Como se demuestra en [18], el uso de un mallado regular garantiza la obtención de los mejores resultados. Por ello, para cada frame se ha definido

un mallado regular 5x5 y los descriptores SIFT 3D serán calculados en los 16 puntos de intersección interiores (ver figura 2.8).

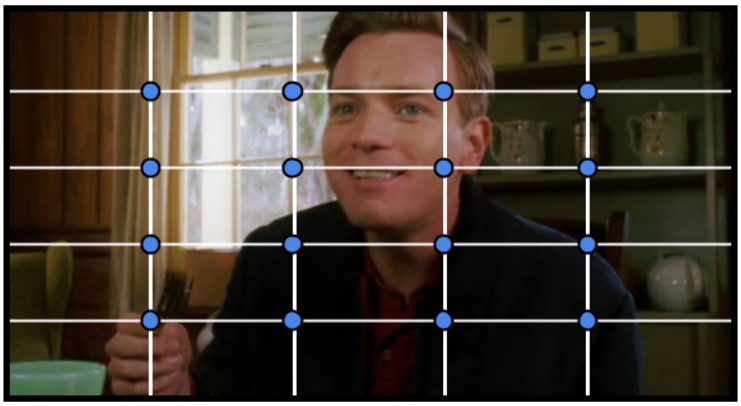

Figura 2.8: Localizaciones para la extracción de los descriptores SIFT 3D en cada frame. Dichas localizaciones son las intersecciones del mallado regular definido.

Para el entrenamiento del vocabulario se han utilizado esta vez tres frames de cada uno de los 144 vídeos que componen nuestro dataset. Dichos frames han sido seleccionados siguiendo el mismo criterio utilizado para los SIFT 2D (figura 2.2). Una vez construída la bolsa de 600 palabras asociada a estos descriptores espacio-temporales, se computan los histogramas en todos los frames del dataset en una escala 2x2 y una escala 4x4, dando lugar a un histograma de descriptores resultante con la misma estructura que el presentado en la ecuación 2.8.

Término binario asociado

La formulación del término binario de la energía definido a partir de los descriptores de bolsa de palabras basados en SIFT 3D es:

$$\Psi^{3D} = \sum_{s,q} \Delta_{s,q}^{BoW^{3D}}(l_s, l_q) \tag{2.16}$$

Con las mismas consideraciones que las hechas para el término binario de bolsa de palabras basado en SIFT 2D, para un dataset de N vídeos, el término binario es de la forma:

$$\Psi = \begin{bmatrix} 0 & \Delta_{1,2}^{BoW^{3D}} & \Delta_{1,3}^{BoW^{3D}} & \cdots & \Delta_{1,N}^{BoW^{3D}} \\ \Delta_{2,1}^{BoW^{3D}} & 0 & \Delta_{2,3}^{BoW^{3D}} & \cdots & \Delta_{2,N}^{BoW^{3D}} \\ \Delta_{3,1}^{BoW^{3D}} & \Delta_{3,2}^{BoW^{3D}} & 0 & \cdots & \Delta_{3,N}^{BoW^{3D}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Delta_{N,1}^{BoW^{3D}} & \Delta_{N,2}^{BoW^{3D}} & \Delta_{N,3}^{BoW^{3D}} & \cdots & 0 \end{bmatrix} \tag{2.17}$$

Término unario asociado

El término unario de la energía definido a partir de los descriptores de bolsa de palabras basados en SIFT 3D es de la forma:

$$\Phi_1^{3D} = \sum_s \Delta_s^{BoW^{3D}}(l_s, \Omega_i) \quad \forall i \in \Big[1, C\Big] \tag{2.18}$$

Para M frames se tiene:

$$\Phi_1^{2D} = \begin{bmatrix} \Delta_1^{BoW^{3D}}(l_1, \Omega_1) & \cdots & \Delta_s^{BoW^{3D}}(l_s, \Omega_1) & \cdots & \Delta_M^{BoW^{3D}}(l_M, \Omega_1) \\ \Delta_1^{BoW^{3D}}(l_1, \Omega_2) & \cdots & \Delta_s^{BoW^{3D}}(l_s, \Omega_2) & \cdots & \Delta_M^{BoW^{3D}}(l_M, \Omega_2) \\ \Delta_1^{BoW^{3D}}(l_1, \Omega_3) & \cdots & \Delta_s^{BoW^{3D}}(l_s, \Omega_3) & \cdots & \Delta_M^{BoW^{3D}}(l_M, \Omega_3) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Delta_1^{BoW^{3D}}(l_1, \Omega_C) & \cdots & \Delta_s^{BoW^{3D}}(l_s, \Omega_C) & \cdots & \Delta_M^{BoW^{3D}}(l_M, \Omega_C) \end{bmatrix} \tag{2.19}$$

2.4. Término basado en la detección de personas

Dado que en todos nuestros vídeos las acciones que se quieren reconocer son llevadas a cabo por seres humanos, la detección de personas será una herramienta de gran importancia en nuestro planteamiento. La información del número de individuos presentes en cada frame será usada para mejorar la formulación de nuestra energía y garantizar mejores resultados.

Hoy en día la detección de personas en imágenes es un problema muy estudiado. De los numerosos métodos existentes, se ha decidido estudiar dos: la detección de caras y la detección de cuerpos. Para la detección de caras se utiliza una herramienta que posee *MATLAB*, basada en el algoritmo de detección de Viola-Jones [19] y en un modelo ya entrenado. El detector de cuerpos usado fue desarrollado por el *CALVIN group* de la ETH de Zurich y sus resultados han sido evaluados en [20].

La utilización de ambos métodos permitirá la obtención de resultados satisfactorios dada la gran variedad de escenas y situaciones que se dan en los vídeos: hay escenas en las que sólo se puede ver la cara de la persona que está llevando a cabo la acción, mientras que en otras es precisamente el rostro la parte del cuerpo que se encuentra oculta al espectador y se requiere entonces una detección del resto del cuerpo del individuo.

Resulta evidente pensar que habrá casos en los que el detector de caras sea más fiable, mientras que en otros, por no ser visible o detectable el rostro de la persona, será el detector de cuerpos el que determine con mayor precisión el número de individuos presentes en la escena. Es necesario, por tanto, determinar el peso de cada una de las dos detecciones (ecuación 2.20) que proporciona los mejores resultados. Este estudio será llevado a cabo en la fase de experimentación.

$$d = \delta_1 d^{\text{Face}} + \delta_2 d^{\text{Body}} \quad con \quad \delta_2 = 1 - \delta_1 \tag{2.20}$$

Tras aplicar el detector de personas en todos los vídeos y conocer el número de individuos que aparecen en cada frame, se está en disposición de definir el término unario de la energía asociado a la detección de personas, que inevitablemente estará condicionado por la precisión del detector diseñado.

$$\Phi_2 = \sum_s D(l_s, d), \quad D = \begin{cases} \omega_0 & \leftarrow \quad d = 0 \\ \omega_1 & \leftarrow \quad d = 1 \\ \omega_2 & \leftarrow \quad d \geq 2 \end{cases} \tag{2.21}$$

Hay tres posibles casos:

- $d = 0$. Si no se han detectado personas es casi seguro que en ese frame no está teniendo lugar acción alguna.
- $d = 1$. Si sólo se ha detectado una persona las acciones más probables deberían ser aquellas que sólo requieren un único individuo para ser llevadas a cabo.
- $d \geq 2$. Si se detectan dos o más personas es probable que nos encontremos ante una escena en la que aparece una multitud. Resulta complicado decidir en este caso cuáles son las acciones más probables, pues se cumplen los requisitos de número de individuos presentes en escena para todas las acciones del dataset.

2.5. Término de acciones centradas

El último de los términos unarios de la energía, que se ha denominado término de acciones centradas, surge para dar respuesta a una realidad presente en los vídeos con los que se ha trabajado y en casi cualquier vídeo realista en general: lo relevante del vídeo, la acción, suele estar temporalmente centrada en él.

Se debe definir el término, por tanto, como una función que refleje la siguiente información:

- Los frames iniciales y finales de un vídeo tienen una mayor probabilidad de no contener ninguna acción relevante. Esta probabilidad es casi nula en los frames centrales.
- Existe una gran probabilidad de que los frames centrales del vídeo contengan la acción buscada. Por el contrario, la probabilidad de que los extremos del vídeo contengan alguna acción es muy reducida.

Para dar respuesta a estas suposiciones se han diseñado dos definiciones para el término de acciones centradas, una triangular y una de tramos exponenciales, las cuales se detallan a continuación.

Definición triangular

Esta definición es, sin duda, la menos flexible de las dos propuestas pues, debido a su linealidad, contempla la posibilidad de que las acciones sean breves y estén profundamente centradas en el vídeo. Intuitivamente, nos permitirá realizar una mejor segmentación de aquellos frames que no contengan acción ya que es muy probable que éstos se encuentren al inicio y al final del vídeo.

$$\Phi_3^{\Delta}(\text{Action}) = \begin{cases} \dfrac{-2k}{n-2}s + \dfrac{nk}{n-2} & \forall s \in \left[1, \frac{n}{2}\right] \\ \dfrac{2k}{n}s - k & \forall s \in \left(\frac{n}{2}, n\right] \end{cases} \tag{2.22}$$

$$\Phi_3^{\Delta}(\text{No Action}) = k - \Phi_3^{\Delta}(\text{Action}) \tag{2.23}$$

Definición exponencial

A diferencia de la triangular, la definición exponencial asigna potenciales altos o bajos (según si se trata de «acción» o «no acción») en regiones de frames muy reducidas al inicio y al final de los vídeos. Ésto nos será de gran utilidad a

la hora de segmentar los frames que contengan acciones relevantes, pues seguro que se encuentran en las regiones centrales del vídeo.

$$\Phi_3^{exp}(\text{Action}) = \begin{cases} \dfrac{ke^{\frac{n}{2}}}{e^{\frac{n}{2}-1}}e^{-s} - \dfrac{k}{e^{\frac{n}{2}-1}-1} & \forall s \in \left[1, \frac{n}{2}\right] \\ \dfrac{k}{e^n - e^{\frac{n}{2}}}e^s - \dfrac{ke^{\frac{n}{2}}}{e^n - e^{\frac{n}{2}}} & \forall s \in \left(\frac{n}{2}, n\right] \end{cases} \tag{2.24}$$

$$\Phi_3^{exp}(\text{No Action}) = k - \Phi_3^{exp}(\text{Action}) \tag{2.25}$$

En la figura 2.9 pueden verse gráficamente las dos definiciones implementadas.

A la vista de las funciones definidas y de los razonamientos expuestos, se puede afirmar que la definición triangular es mejor segmentando los frames sin acción y la exponencial los frames que sí contienen alguna de las acciones de estudio. Por ello se usarán ambas definiciones para configurar el término unario de acciones centradas:

$$\Phi_3 = \epsilon_1 \Phi_3^{exp} + \epsilon_2 \Phi_3^{\Delta} \quad \text{con} \quad \epsilon_2 = 1 - \epsilon_1 \tag{2.26}$$

2.6. Minimización de la energía

Como se vio en 2.1, minimizar nuestra función de energía, $E(L|\alpha)$, supone encontrar el etiquetado, L^*, y el conjunto de parámetros, α^*, que hacen que el coste sea mínimo y, por tanto, el reconocimiento sea óptimo.

$$L^* \leftarrow \underset{L}{\operatorname{argmin}}\Big(E(L|\alpha)\Big) \tag{2.27}$$

$$\alpha^* \leftarrow \underset{\alpha}{\operatorname{argmin}}\Big(E(L|\alpha)\Big) \tag{2.28}$$

En [21] se explica cómo funciones de energía de la forma

$$E(f) = \sum_{p \in P} D_p\Big(f_p\Big) + \sum_{p,q \in N} V_{p,q}\Big(f_p, f_q\Big), \quad N \in P\text{x}P \tag{2.29}$$

son en general difíciles de minimizar, pues son funciones no convexas con miles de dimensiones. Nuestra energía, como se ha visto en el presente capítulo, responde a la forma general descrita por la ecuación 2.29. En los últimos años se han desarrollado algoritmos eficientes basados en *graph cuts* para minimizar estas funciones de energía. Aplican el teorema «*max-flow, min-cut*» para resolver los grafos construídos.

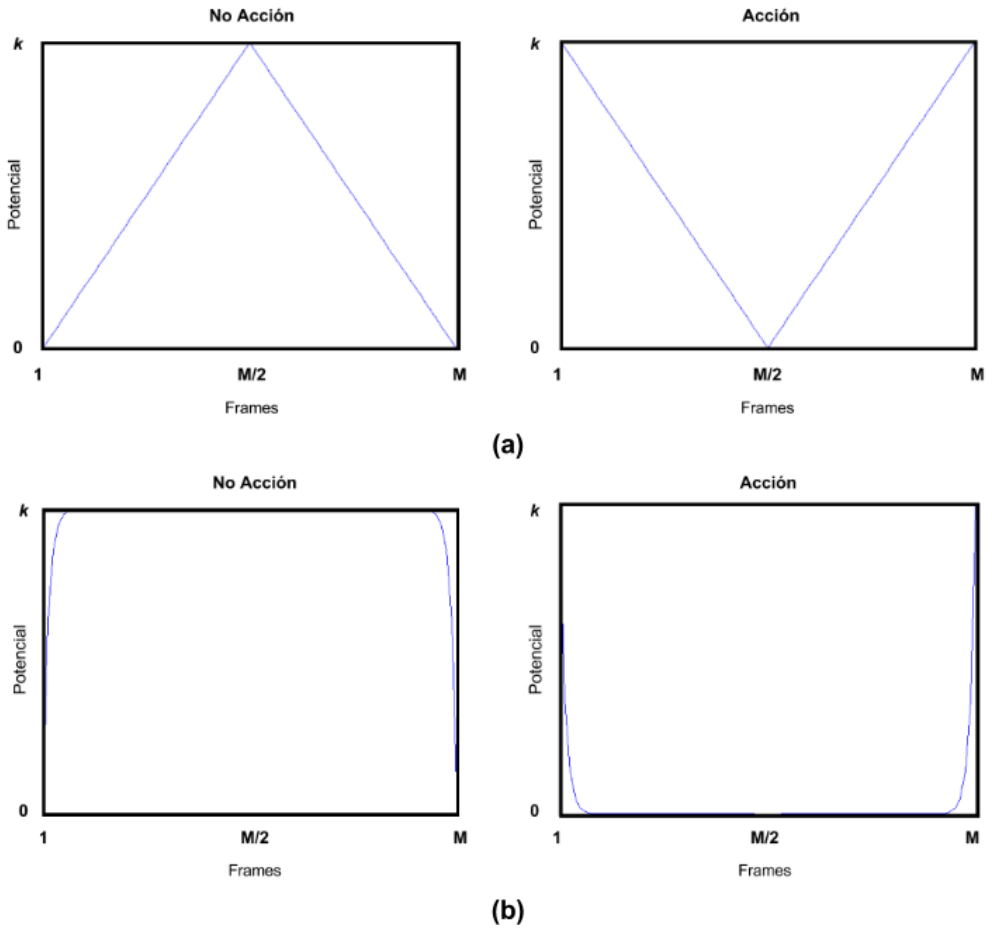

Figura 2.9: Definiciones triangular (a) y exponencial (b) del término unario de acciones centradas. Modelan las probabilidades de que una acción se halle en el centro del vídeo y que los frames iniciales y finales del mismo no contengan ninguna acción.

En nuestro problema, el grafo está formado por M nodos, que se corresponden con los frames de los vídeos del dataset. Encontrar el corte mínimo supone asignar a cada nodo la etiqueta (de las 13 posibles: *None, AnswerPhone, Drivecar, Eat, FightPerson, GetOutCar, HandShake, HughPerson, Kiss, Run, SitDown, SitUp* y *StandUp*) que hace que el flujo sea máximo. La figura 2.10 recoge la estructura de nuestro grafo.

La función de minimización implementada por el algoritmo de *graph cuts* utilizado [22, 23] es de la forma:

$$\Big[L, E', E\Big] = GC\Big(L_o, \Psi, \Phi, \Gamma, \beta\Big) \tag{2.30}$$

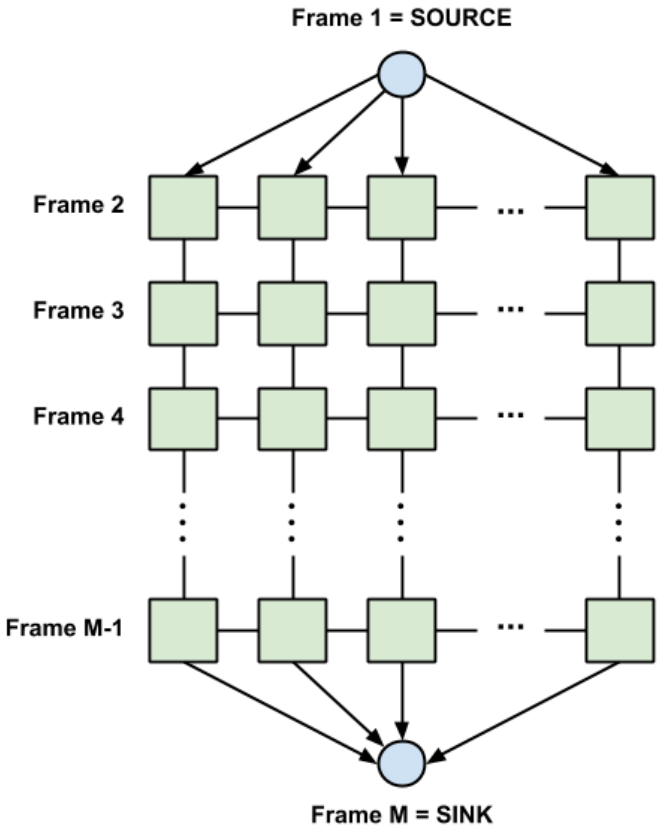

Figura 2.10: Estructura del grafo de nuestro problema. Las M filas representan los M nodos o frames del dataset y las columnas son todas las posibles combinaciones de etiquetas.

donde los parámetros de entrada son:

- L_o: etiquetado inicial de los nodos del grafo.
- Ψ, Φ: términos binario y unario de la energía.
- Γ: matriz de dimensiones CxC que especifica el coste de cambio de etiqueta para cada nodo adyacente en el grafo. Permite, por ejemplo, hacer que el coste de cambiar de la etiqueta *DriveCar* a la etiqueta *GetOutCar* sea pequeño y que el coste de cambiar de *DriveCar* a *FightPerson* sea alto.
- β: parámetro de selección del método de minimización (0-*swap*, 1-*expansion*).

La función determina:

- E: energía asociada etiquetado inicial.
- E': energía resultante tras la minimización. Se debe cumplir siempre que $E' < E$.
- L^*: etiquetado óptimo.

En la figura 2.11 se puede ver un esquema de los parámetros de entrada y salida.

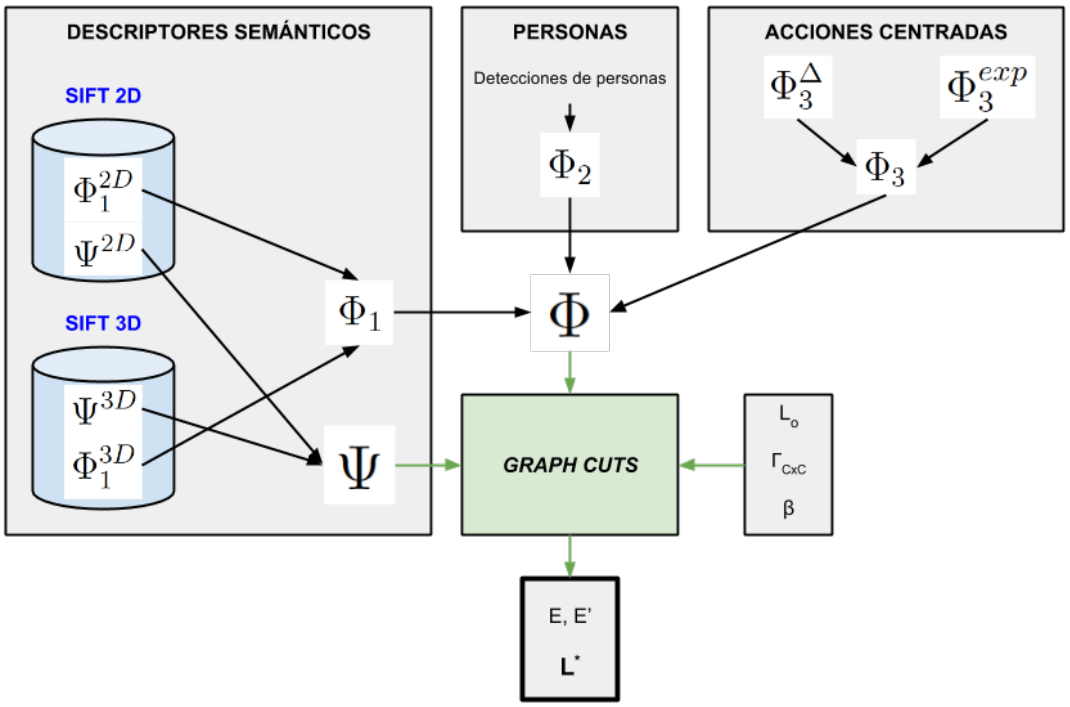

Figura 2.11: Esquema de la función que implementa los *graph cuts*.

Capítulo 3

Resultados experimentales

3.1. Introducción

A la hora de resolver un problema de visión por computador mediante la minimización de una función de coste la experimentación juega un papel crucial. Es necesario llevar a cabo constantes ensayos durante el proceso de formulación de la energía, pues es la única forma de verificar si la adición de nuevos términos lleva a unos resultados más satisfactorios o no. Además, dichos ensayos servirán para configurar de forma óptima esos términos de la energía a través de sus parámetros.

Los ensayos serán realizados en un dataset de vídeos creado expresamente para este trabajo. Dichos vídeos han sido cuidadosamente anotados para poder definir un *ground truth* con el que evaluar los resultados de la minimización.

Tras analizar la creación del dataset y el *ground truth*, se detallará exhaustivamente el proceso de parametrización y configuración llevado a cabo para ajustar la ecuación de energía que modela nuestro problema. Por último, se presentarán y analizarán los resultados obtenidos al minimizar la energía resultante, tanto en nuestro dataset como en el dataset de acciones KTH [3].

3.2. Dataset y *ground truth*

Uno de los principales objetivos que se marcaron al inicio del presente trabajo fue la creación de un método débilmente supervisado que fuera capaz de llevar a cabo la segmentación temporal y el reconocimiento de acciones en vídeos realista que supusieran un verdadero reto. El método debía ser capaz de funcionar, pues, en lo que podríamos llamar un entorno de vídeos realistas. El núcleo del algoritmo debía desconocer las acciones a identificar y éstas debían ser llevadas a cabo de formas diversas en entornos variados. Qué mejor para

ello que utilizar escenas de películas: diversidad de individuos, de escenarios, de iluminación, de *frame rate*...

Por este motivo se decidió crear nuestro dataset a partir del dataset ya existente llamado *Hollywood2* [16]. Los 144 vídeos seleccionados se agrupan en 12 categorías, o acciones, diferentes: *AnswerPhone, Drivecar, Eat, FightPerson, GetOutCar, HandShake, HughPerson, Kiss, Run, SitDown, SitUp* y *StandUp*. Cada una de ellas consta de 12 vídeos. En la figura 3.1 se presenta un ejemplo de cada acción.

A la hora de elegir los vídeos se ha seguido únicamente el criterio previamente expuesto: gran variedad en todos los aspectos con el objetivo de crear un reto para el algoritmo.

Figura 3.1: Acciones contenidas en nuestro dataset.

Cuando los experimentos llevados a cabo sean analizados se harán frecuentes referencias a un «dataset reducido». Dicho dataset, que se compone de 6 categorías (las 6 primeras de las antes mencionadas) y 6 vídeos por categoría, ha sido utilizado en la fase de ajuste de la energía para determinar la configuración de parámetros que conduce a los resultados más satisfactorios.

Como ya se ha mencionado, la energía formulada se evaluará también en el dataset de acciones KTH. Este dataset se compone de 6 acciones llevadas a cabo en 4 escenarios diferentes, que pueden verse en la figura 3.2. Se han seleccionado 6 vídeos para cada una de ellas.

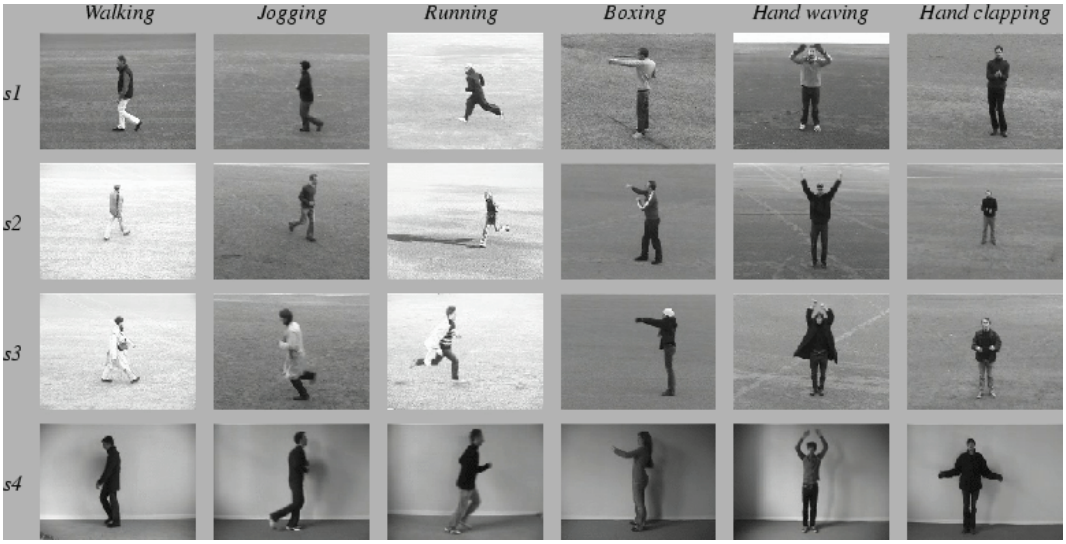

Figura 3.2: Acciones y escenarios contenidos en el dataset KTH.

Una vez se ha elaborado el dataset, es necesario definir un criterio en base al cuál determinar la bondad de los resultados obtenidos en las simulaciones. Para ello, los vídeos han sido anotados utilizando el software *Virtualdub* [1], que permite procesarlos frame a frame. De esta forma se pueden determinar los frames «exactos» de inicio y final de las acciones. Estos frames han sido utilizados para crear el *ground truth* con el que se evaluarán los resultados obtenidos en la minimización, de la forma que se expone a continuación.

El número de frames correctamente etiquetados en la minimización, L_{ok}, sobre el total de frames del dataset, M, determinará la precisión de la segmentación:

$$\text{Precisión en la segmentación} \equiv \%S_{OK} = \frac{L_{ok}}{M}$$

[1] http://www.virtualdub.org/

Se definen, además, $\%S_{OK}^{Action}$ como la precisión en la segmentación de aquellos frames que contienen acción y $\%S_{OK}^{None}$ como la precisión en la segmentación de frames en los que no hay acción.

El reconocimiento de las acciones, por su parte, dependerá del número de vídeos cuya acción ha sido correctamente reconocida, N_{ok}, sobre el total de vídeos del dataset, N:

$$\text{Precisión en el reconocimiento} \equiv \%R_{OK} = \frac{N_{ok}}{N}$$

3.3. Ajuste de los parámetros

Con la finalidad de encontrar la configuración de parámetros óptima se han diseñado una serie de experimentos que serán llevados a cabo en el dataset reducido. Estos experimentos conforman por tanto el proceso de ajuste paramétrico de la energía. Se debe tener en cuenta que dicho proceso es altamente iterativo y se ha realizado de forma paralela a la formulación teórica.

Para decidir la configuración óptima de parametros, en cada experimento se tendrá en cuenta únicamente la precisión en la segmentación, $\%S_{OK}$. Ésto se debe a que nuestro reconocimiento está fuertemente ligado a la segmentación: la acción asociada a cada vídeo se obtiene mediante el cálculo de la etiqueta distinta de *None* que aparece con mayor frecuencia en sus frames.

Se presentan a continuación los resultados más relevantes de los experimentos realizados.

La norma euclídea frente a la distancia de Bhattacharyya

En un primer momento se pensó en trabajar con una ecuación de energía definida únicamente mediante los términos derivados de los descriptores semánticos. Para ello, era necesario saber cuál de los dos métodos propuestos para la comparación de los histogramas ofrecía los mejores resultados: la norma euclídea de la diferencia de histogramas o la distancia de Bhattacharyya existente entre ellos.

En la tabla 3.1 se puede ver cómo el uso de la norma euclídea para la comparación de descriptores es más preciso que la distancia de Bhattacharyya. Si bien los resultados para los frames que no contienen ninguna acción de las contempladas son idénticos, la norma euclídea es un 7 % mejor para detectar aquellos frames que sí contienen acción.

Los histogramas de descriptores, como se pudo ver en la ecuación 2.8, no son otra cosa que vectores. Teóricamente, la distancia de Bhattacharyya debería aportar una información más rica a la hora de comparar dos vectores. Los

Método	% S_{Ok}	% S_{Ok}^{None}	% S_{Ok}^{Action}
Norma euclídea	34.12	15.41	41.03
Distancia de Bhattacharyya	29.00	15.41	34.02
Distancia de Bhattacharyya*	33.87	26.73	36.51

Tabla 3.1: Resultados de la utilización de la norma euclídea y la distancia de Bhattacharyya para la comparación de los histogramas. Se puede ver cómo los resultados del nuevo método de aplicación de la distancia de Bhatacharyya (indicado en la tabla con *) son mejores que los obtenidos con el método inicial. A pesar de ello, los resultados obtenidos con la norma euclídea siguen siendo superiores. Estos resultados son idénticos para ambos métodos de aplicación.

resultados obtenidos, que contradicen claramente esta suposición, se deben a que el cálculo de la distancia de Bhattacharyya está considerando los histogramas como vectores de 12000 componentes, donde todas ellas representan diferentes resultados de un experimento, y ésto no es cierto. Los vectores se han construído como la concatenación de los histogramas computados al aplicar el vocabulario en 20 regiones diferentes de la imagen (4 para la escala 2x2 y 16 para la escala 4x4), por lo que están formados por 20 histogramas diferentes e independientes.

Se puede considerar en este punto la aplicación de la distancia de Bhattacharyya de forma independiente en cada uno de esos 20 histogramas que forman el histograma de descriptores total.

$$\Delta_{s,q}^{BoW} = \omega_{s,q} = \|\gamma\|, \quad \gamma = \Big(db(h_{s_1}, h_{q_1}), db(h_{s_2}, h_{q_2}), ..., db(h_{s_{20}}, h_{q_{20}})\Big) \quad (3.1)$$

Para que el experimento sea coherente la norma euclídea deberá ser aplicada de la misma forma.

$$\Delta_{s,q}^{BoW} = \omega_{s,q} = \|\gamma\|, \quad \gamma = \Big(\|\Delta_{h_{s_1},h_{q_1}}\|, \|\Delta_{h_{s_2},h_{q_2}}\|, ..., \|\Delta_{h_{s_{20}},h_{q_{20}}}\|\Big) \quad (3.2)$$

Los resultados se recogen también en la tabla 3.1. A la vista de ellos y teniendo en cuenta la configuración de nuestros descriptores de bolsa de palabras, los potenciales asociados a las diferencias entre los histogramas de descriptores de dos frames, $\omega_{s,q}$, serán calculados mediante la norma euclídea.

Términos de bolsa de palabras

En el proyecto se han implementado dos tipos de términos de bolsa de palabras, uno basado en descriptores SIFT 2D y otro basado en descriptores

SIFT 3D y es necesario, por tanto, determinar el peso relativo óptimo entre ambos. Esta parametrización se ha dividido en dos fases:

- **Fase I: parametrización del término binario de la energía**. En 2.2, utilizando la terminología del cálculo variacional, se indicaba que el término binario podía ser considerado un regularizador. En nuestro caso establece los costes relativos de unión entre dos frames del problema. Por ello se busca en primer lugar, para una configuración fija del término de datos (unario), el peso relativo óptimo entre la componente del término binario basada en SIFT 2D y la basada en SIFT 3D. Los resultados demuestran que, al ser costes relativos entre nodos y computarse de forma masiva, la minimización de la energía es independiente de la parametrización del término binario siempre y cuando se mantenga esa relatividad. Ésto es, siempre que se calcule el término utilizando el mismo criterio.

  Sean

  $$L_a^* \leftarrow \underset{L}{\operatorname{argmin}}\Big(E(L|\alpha)\Big) \quad \text{con} \quad \Big\{\lambda_1^a, \lambda_2^a\Big\} \tag{3.3}$$

  $$L_b^* \leftarrow \underset{L}{\operatorname{argmin}}\Big(E(L|\alpha)\Big) \quad \text{con} \quad \Big\{\lambda_1^b, \lambda_2^b\Big\} \tag{3.4}$$

  Se puede concluir, en base a los experimentos llevados a cabo, que

  $$L_a^* = L_b^* \quad \forall\Big\{\lambda_1, \lambda_2\Big\} \subset \alpha \tag{3.5}$$

- **Fase II: parametrización del término unario de bolsa de palabras**. Conocida la configuración óptima del regularizador, se determina el peso relativo entre las componentes del término unario de bolsa de palabras que conduce a los mejores resultados. Los resultados se recogen en la tabla 3.2.

Definición del término basado en la detección de personas

El primer paso a seguir en la detección de personas es el ajuste del detector, es decir, asignar un peso relativo a la detección de caras y a la detección de cuerpos. Para ello, se ha llevado a cabo un estudio de ambas detecciones en 144 frames extraídos de los vídeos de nuestro dataset. Concretamente, se trata de los frames centrales de cada uno de los vídeos. Se han elegido estas imágenes porque la probabilidad de que en el frame central de cada vídeo tenga lugar la acción es muy elevada. En la tabla 3.3 se presentan los resultados del estudio, mientras que en la figura 3.3 se recogen unos ejemplos de la detección de personas en nuestras imágenes.

$\xi 1$	$\xi 2$	% S_{Ok}	% S_{Ok}^{Action}
0	1	25.91	29.78
0.05	0.95	33.21	39.78
0.10	0.90	32.53	38.84
0.15	0.85	32.57	38.90
0.20	0.80	32.90	39.36
0.25	0.75	33.56	40.26
0.30	0.70	33.55	40.24
0.35	0.65	33.87	40.68
0.40	0.60	34.02	40.89
0.45	0.55	34.14	41.05
0.50	0.50	34.26	41.22
0.55	0.45	34.26	41.22
0.60	**0.40**	**34.33**	**41.31**
0.65	0.35	34.30	41.28
0.70	0.30	34.23	41.18
0.75	0.25	34.16	41.08
0.80	0.20	34.22	41.16
0.85	0.15	34.14	41.05
0.90	0.10	34.09	40.99
0.95	0.05	34.11	41.01
1	0	34.12	41.03

Tabla 3.2: Resultados de la parametrización del término unario de bolsa de palabras. La componente basada en SIFT 3D mejora ligeramente los resultados de la segmentación. Se marcan en negrita los valores seleccionados.

Detector	**Personas reconocidas (%)**	**Falsos positivos**	**Precisión etiquetado (%)**
Caras	18.07	10	85.29
Cuerpos	22.74	27	73.00

Tabla 3.3: Principales resultados del estudio comparativo entre el detector de caras y el detector de cuerpos. La precisión en el etiquetado representa el ratio de acierto teniendo en cuenta únicamente las etiquetas introducidas por los detectores, es decir, aquellas regiones de la imagen marcadas como caras o cuerpos.

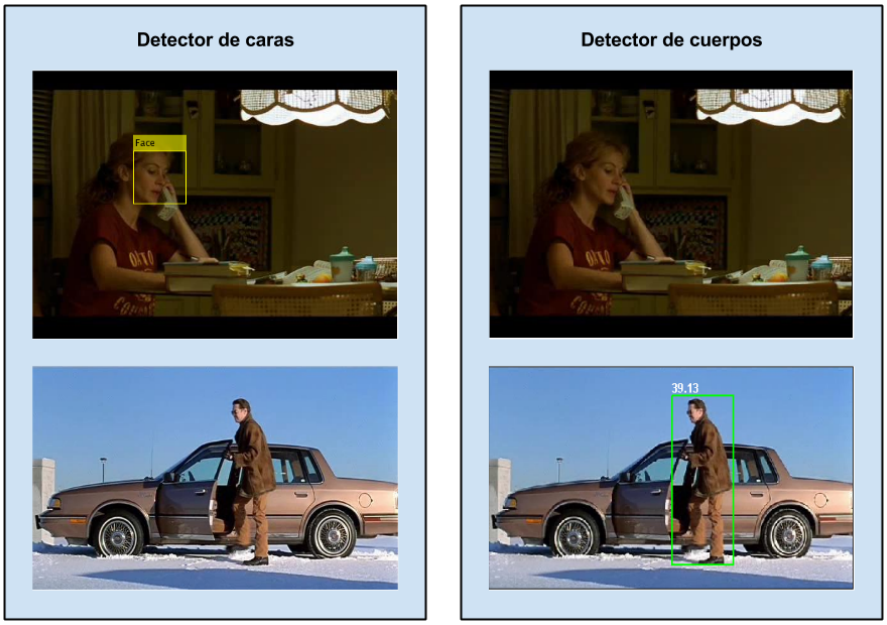

Figura 3.3: Ejemplos de detección de personas en los vídeos. Cuando el cuerpo de la persona que se quiere detectar no aparece por completo en la imagen es el detector de caras el que presenta una precisión mayor. Por su parte, el detector de cuerpos se impone a la hora de detectar individuos que aparecen alejados en la imagen.

Si bien se ha observado que el detector de cuerpos reconoce un mayor número de personas en las imágenes de prueba, el elevado número de falsos positivos introducidos hace que su precisión en el etiquetado sea menor que la del detector de caras.

Ante la imposibilidad de determinar el mejor de los dos descriptores y al ser ambos muy útiles según las características del frame a tratar, se ha optado por dar el mismo peso a ambos, con lo que se tiene:

$$\epsilon_1 = \epsilon_2 \Rightarrow d = \frac{1}{2} d^{\text{Face}} + \frac{1}{2} d^{\text{Body}} \tag{3.6}$$

Una vez diseñado el detector que se va a utilizar se debe determinar la configuración óptima del término unario basado en la detección de personas. Es necesario encontrar, por tanto, la mejor combinación de potenciales asignados en función del número de individuos detectados. Para facilitar esta labor, se han dividido las acciones de estudio en 3 grupos:

- Sin acción, acc_0.

- Acciones individuales, acc_1; acciones que pueden ser efectuadas por una sola persona. Son: *AnswerPhone, Drivecar, Eat, GetOutCar, Run, SitDown, SitUp* y *StandUp*.

- Acciones colectivas, acc_2; acciones que requieren la interacción de dos o más individuos para poder ser llevadas a cabo. Son las restantes: *FightPerson, HandShake, HugPerson* y *Kiss*.

Esta división propuesta no es, ni mucho menos, absoluta. Hay acciones, como por ejemplo *GetOutCar* o *Run*, que pueden ser realizadas por una sola persona, pero es posible que existan escenas donde varios individuos salgan de sus coches o aperezcan multitudes corriendo.

Por otra parte, es posible que en las escenas se detecten personas que no están llevando a cabo ninguna acción. Pensemos, por ejemplo, en un abrazo que tiene lugar en unas oficinas donde hay gente trabajando; son dos los individuos que realizan la acción de interés, pero es posible que el detector de personas haya encontrado muchas más en la escena. También puede ocurrir que en esa misma oficina ocurra la acción *AnswerPhone*; dicha acción es realizada por una persona, pero el número de individuos en la escena es mucho mayor.

La conclusión de todo esto es que la detección de un número determinado de personas en una escena no es un factor determinante a la hora de decidir qué acción se está llevando a cabo. Ningún detector de personas es perfecto, y el nuestro no es una excepción. Sin embargo, estamos en disposición de asignar probabilidades, y por consiguiente potenciales, a aquellas acciones que se consideren más probables para un número de personas detectadas dado. Se debe asumir, por último, que el error en la detección va a estar siempre presente.

Ha quedado claro que conseguir una combinación de potenciales perfecta para definir el término basado en la detección de personas es imposible. Se debe llegar a un compromiso entre lo que es más probable que esté ocurriendo y la precisión de nuestro detector. Así pues, se plantea la asignación de potenciales como sigue:

$$d = 0 \Rightarrow \begin{cases} \omega_0(acc_0) \rightarrow P(acc_0 \mid f = 0) \approx 1 \Rightarrow \omega_0(acc_0) = 0 \\ \\ \omega_0(acc_{1,2}) \rightarrow P(acc_{1,2} \mid f = 0) \approx 0 \Rightarrow \omega_0(acc_{1,2}) = 1 \end{cases} \tag{3.7}$$

$$d = 1 \Rightarrow \begin{cases} \omega_1(acc_0) = K_- \\ \\ \omega_1(acc_1) = K_+ \\ \\ \omega_1(acc_2) = K_- \end{cases} \tag{3.8}$$

K_+	K_-	% S_{Ok}	% S_{Ok}^{Action}
0	{0.20, 0.25, 0.30, 0.35, 0.50, 0.80}	42.81	40.99
0.05	{0.20, 0.25, 0.30, 0.35, 0.50, 0.80}	42.81	40.99
0.10	{0.20, 0.25, 0.30, 0.35, **0.50**, 0.80}	**42.81**	**40.99**
0.15	0.20	42.73	40.87
0.15	{0.25, 0.30, 0.35, 0.50, 0.80}	42.81	40.99
0.20	0.20	39.69	36.72
0.20	0.25	42.73	40.87
0.20	{0.30, 0.35, 0.50, 0.80}	42.81	40.99

Tabla 3.4: Resultados de los experimentos realizados para determinar el valor óptimo de los potenciales que definen el término unario basado en la detección de personas. Se marcan en negrita los valores seleccionados.

$$d \geq 2 \Rightarrow \begin{cases} \omega_2(acc_0) = K_- \\ \omega_2(acc_1) = K_+ \\ \omega_2(acc_2) = K_+ \end{cases} \tag{3.9}$$

Con el fin de cuantificar la probabilidad y asignar los potenciales se ha usado la siguiente notación: $K_+ \equiv$ *Probable*, y $K_- \equiv$ *Menos probable*. De esta forma es sencillo simular el término hasta encontrar los valores óptimos para estos potenciales. En la tabla 3.4 se pueden ver los resultados de los experimentos llevados a cabo a tal efecto.

Parametrización del término de acciones centradas

El último término unario servirá para «afinar» la segmentación temporal y el reconocimiento de las acciones en los vídeos.

Como se describía en 2.5, mediante los parámetros ϵ_1 y ϵ_2 se determinará la relación de pesos óptima entre la componente exponencial y la componente triangular del término. Con ello se buscará el mejor compromiso entre acierto en la segmentación total (frames que contienen acción y frames que no la contienen) y acierto en la segmentación de acciones, pues ésto conducirá a un mejor reconocimiento de las mismas.

La tabla 3.5 recoge los resultados obtenidos en simulación.

ϵ_1	ϵ_2	% S_{Ok}	% S_{Ok}^{None}	% S_{Ok}^{Action}
0	1	**42.14**	**48.37**	39.84
0.05	0.95	41.37	44.63	40.16
0.10	0.90	41.23	43.02	40.57
0.15	0.85	41.35	42.24	41.03
0.20	0.80	41.49	41.36	41.54
0.25	0.75	41.66	40.48	42.10
0.30	0.70	41.91	39.65	42.75
0.35	**0.65**	**42.11**	**37.73**	**43.73**
0.40	0.60	42.05	35.81	44.36
0.45	0.55	41.66	32.85	44.91
0.50	0.50	40.53	27.45	45.36
0.55	0.45	38.78	20.60	45.49
0.60	0.40	38.18	18.21	**45.55**
0.65	0.35	37.70	16.45	45.55
0.70	0.30	37.51	15.72	45.55
0.75	0.25	37.02	13.91	45.55
0.80	0.20	36.23	11.00	45.55
0.85	0.15	36.08	10.43	45.55
0.90	0.10	35.74	9.19	45.55
0.95	0.05	35.67	8.93	45.55
1	0	35.16	7.01	45.55

Tabla 3.5: Resultados de la parametrización de las dos componentes del término unario de acciones centradas, donde ϵ_1 es el parámetro de la componente exponencial y ϵ_2 el parámetro de la componente triangular. Se han resaltado en negrita los valores de interés. Se ve que utilizando únicamente la definición triangular ($\epsilon_1 = 0$), se obtiene el mejor resultado en segmentación total y en segmentación de frames sin acción. Además, se comprueba que el mejor resultado en la segmentación de frames que sí contienen acción se da para $\epsilon_1 = 0{,}6$ y $\epsilon_2 = 0{,}4$. Con el objetivo de reflejar el compromiso entre segmentación global y de acciones, sin renunciar a una segmentación decente de frames sin acción, se han elegido los valores de los parámetros sombreados: $\epsilon_1 = 0{,}35$ y $\epsilon_2 = 0{,}65$.

3.4. Análisis de resultados

En la sección anterior se ha desarrollado el proceso de ajuste de parámetros llevado a cabo. La función de energía, ya parametrizada, es de la forma:

$$E\big(L|\alpha\big) = \underbrace{\underbrace{0{,}5\Psi^{2D} + 0{,}6\Phi_1^{2D}}_{BoW^{2D}} + \underbrace{0{,}5\Psi^{3D} + 0{,}4\Phi_1^{3D}}_{BoW^{3D}}}_{BoW} + \underbrace{\Phi_2}_{People} + \underbrace{0{,}35\Phi_3^{exp} + 0{,}65\Phi_3^{\Delta}}_{AC} \tag{3.10}$$

Como se explicó, el ajuste de los parámetros se ha llevado a cabo en un dataset reducido, formado por 36 vídeos seleccionados de entre 6 categorías diferentes de nuestro dataset. Es de esperar, por tanto, que los resultados obtenidos al evaluar la energía en el dataset completo difieran de los obtenidos en la parametrización.

Se analizan a continuación los resultados de la minimización en ambos datasets estudiados.

Dataset *KTH*

La precisión en la segmentación temporal y el reconocimiento de acciones, como se puede ver en la tabla 3.6, es mayor en el dataset *KTH*. Era de esperar, pues es un dataset mucho más sencillo en el que hay poca variedad de acciones y escenarios.

Utilizando una función de energía definida a partir de los términos unarios y binarios de bolsa de palabras se ha alcanzado un 38.49 % de acierto en la segmentación temporal de las acciones y un 50 % de acierto en el reconocimiento de las mismas. Que se haya reconocido la acción de forma satisfactoria en uno de cada dos vídeos, utilizando únicamente un vídeo anotado por acción, demuestra que el reconocimiento de acciones débilmente supervisado es posible.

En las matrices de confusión asociadas a los experimentos en el dataset *KTH* (figura 3.4) se observa cómo los descriptores basados en SIFT 2D son más eficaces a la hora de reconocer ciertas acciones, mientras que los basados en SIFT 3D lo son con otras. Es la combinación de ambos la que ha proporcionado los mejores resultados.

Por último, se debe indicar que no ha sido posible estudiar en este dataset el efecto de los términos de la energía basados en la detección de personas y en las acciones centradas que han sido definidos en este proyecto. Ésto se debe a que los vídeos solo contienen acciones (no hay frames sin acción) y, evidentemente, éstas son llevadas a cabo por personas, por lo que la detección de individuos es irrelevante.

Dataset	**Método**	% S_{Ok}	% R_{Ok}
KTH	BoW^{2D}	38.40	47.22
KTH	BoW^{3D}	35.80	50.00
KTH	BoW	38.49	50.00
Hollywood2	$BoW + People + AC$	31.19	33.33

Tabla 3.6: Resultados de la minimización de la función de energía creada en este proyecto para el dataset KTH y el dataset propio creado a partir del *Hollywood2*.

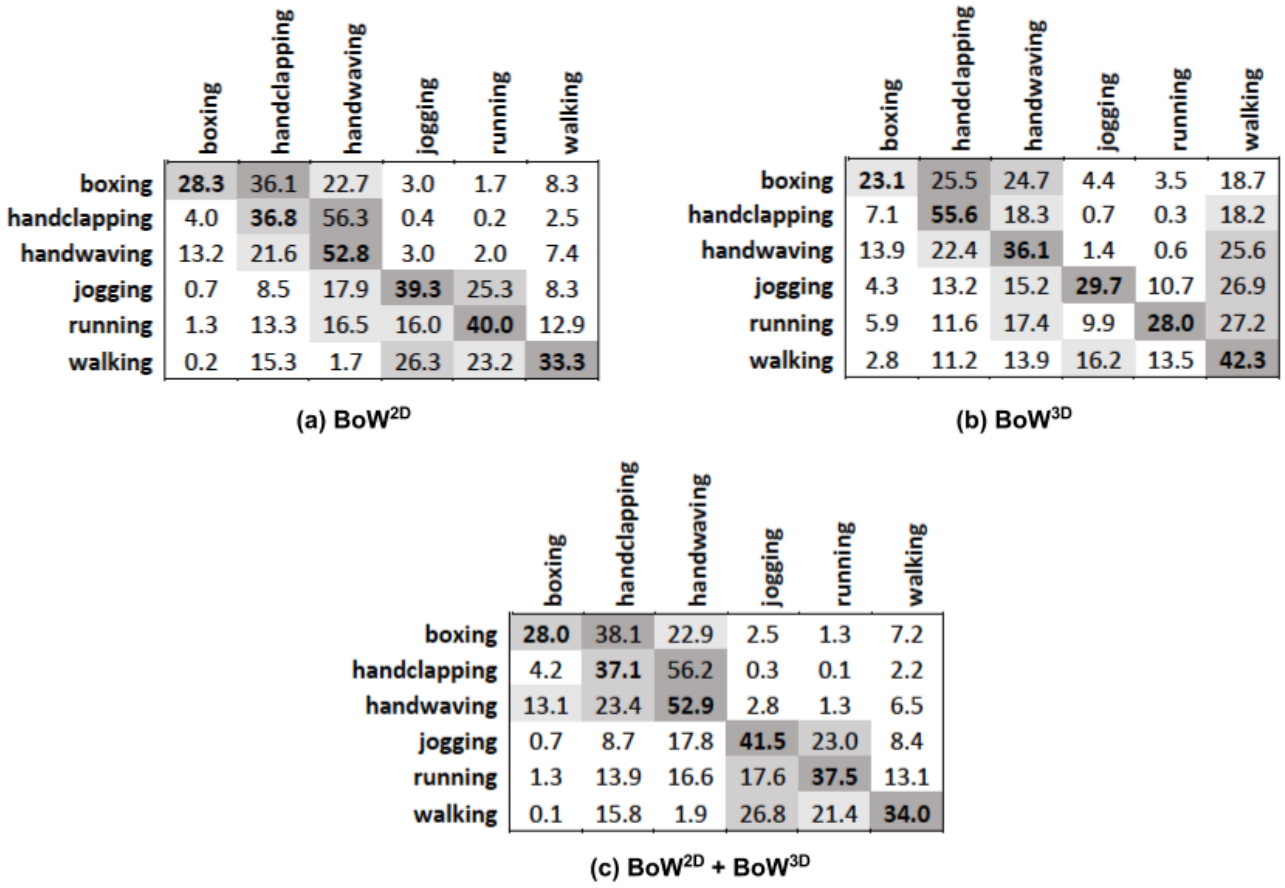

Figura 3.4: Matrices de confusión de los diferentes experimentos llevados a cabo en el dataset *KTH*. (a) es la matriz de confusión asociada al experimento realizado únicamente con descriptores de bolsa de palabras basados en SIFT 2D, (b) la asociada al experimento con descriptores de palabras basados en SIFT 3D y (c) la asociada a la combinación de ambos.

Dataset *Hollywood2*

El dataset creado para el proyecto contiene el doble de acciones que el *KTH* y ésta es una de las razones por las que los resultados de la minimización de la función de energía son peores: 31.19 % de precisión en la segmentación y 33.33 % de acierto en el reconocimiento de acciones. Al haber más acciones se pueden dar muchos más casos de confusión de una acción con otra. Ésto se observa muy bien en la matriz de confusión recogida en la figura 3.5.

	None	AnswerPhone	DriveCar	Eat	FightPerson	GetOutCar	HandShake	HughPerson	Kiss	Run	SitDown	SitUp	StandUp
None	**41.8**	7.9	5.7	1.9	5.5	6.6	2.6	2.2	4.1	3.3	6.6	4.6	7.5
AnswerPhone	12.3	**36.3**	2.2	3.3	2.2	2.4	16.4	7.2	3.5	3.3	4.8	2.0	4.2
DriveCar	7.5	10.4	**28.8**	6.9	3.3	3.7	6.6	1.2	1.1	2.0	15.4	7.4	5.9
Eat	13.9	3.9	1.3	**42.9**	1.9	1.1	4.7	2.4	0.0	6.7	17.7	0.1	3.5
FightPerson	15.1	11.5	10.4	3.9	**24.6**	3.0	6.2	1.0	1.1	2.1	3.8	11.1	6.2
GetOutCar	12.3	2.8	1.0	2.4	3.7	**25.7**	21.9	3.6	3.6	6.8	11.0	2.3	2.7
HandShake	3.1	25.2	11.9	0.0	9.2	0.2	**23.2**	7.0	0.0	2.4	2.9	6.1	8.7
HughPerson	15.8	2.9	12.3	1.1	3.3	0.0	16.6	**20.9**	1.7	2.1	16.3	5.2	2.0
Kiss	5.1	4.3	2.1	7.4	6.5	0.0	0.0	1.6	**49.9**	1.0	2.6	12.9	6.7
Run	17.7	5.3	4.3	11.3	4.3	0.1	11.2	1.6	5.0	**30.8**	2.2	3.0	3.1
SitDown	12.8	11.2	4.5	0.8	1.0	3.9	5.7	0.8	0.0	0.2	**29.0**	21.5	8.7
SitUp	16.6	3.6	3.6	8.6	17.7	0.0	5.0	1.2	0.0	13.1	2.5	**27.2**	0.9
StandUp	12.5	10.8	5.6	6.7	3.2	0.0	11.8	0.3	0.3	16.9	4.0	3.5	**24.6**

Figura 3.5: Matriz de confusión para el dataset creado en el proyecto. Se puede observar cómo para todas las acciones estudiadas salvo *HandShake* el valor más alto se encuentra en la diagonal de la matriz. Destacan, por encima del resto, los valores del reconocimiento de las acciones *None*, *Answerphone*, *Eat* y *Kiss*.

Si tenemos en cuenta que éstos resultados se han obtenido utilizando únicamente un vídeo anotado por acción, frente a los cientos de vídeos anotados que suelen usar los métodos supervisados, los resultados son bastante similares. Si se dispone, por ejemplo, de 30 vídeos anotados para cada acción, es posible realizar comparaciones mucho más precisas para todos los vídeos estudiados, lo que permite obtener mejores resultados.

En la figura 3.6 se han recogido algunos ejemplos del reconocimiento de acciones en el dataset *Hollywood2*.

Figura 3.6: Algunos ejemplos de los resultados del reconocimiento de acciones llevado a cabo en el dataset *Hollywood2*.

Capítulo 4

Conclusiones y líneas futuras

4.1. Conclusiones

El principal objetivo de este proyecto ha sido la creación de un método ligeramente supervisado para la segmentación y el reconocimiento de acciones en vídeos.

Para ello, se ha formulado el problema a través de una ecuación de energía o coste. Dicha energía se compone de términos binarios, referidos a dos frames cualesquiera, y términos unarios, referidos a un sólo frame. Éstos términos se dividen en tres grupos:

- Términos asociados a descriptores de bolsa de palabras (o descriptores semánticos, como se han denominado en este proyecto) basados en algoritmos SIFT 2D y SIFT 3D. Estos descriptores han sido entrenados utilizando un volumen de información anotada mucho menor que el usado otros métodos existentes en la bibliografía. Éste aspecto aleja nuestro método de aquéllos y hace que sea débilmente supervisado.

- Un término basado en la detección de personas en los vídeos. Dicha detección es llevada a cabo, de forma conjunta, por un detector de caras y un detector de cuerpos.

- Términos llamados en el proyecto «de acciones centradas», los cuales modelan la probabilidad de que aparezca una acción en un vídeo en función del instante temporal en que nos encontramos.

El peso relativo entre estos términos de la energía ha sido modelado a través de un conjunto de parámetros, cuyo valor óptimo ha sido determinado mediante experimentación. Para llevarla a cabo se ha creado un dataset de vídeos realistas extraídos de películas. Los vídeos que lo componen han sido

extraídos del dataset ya existente llamado *Hollywood2*. De esta forma hemos conseguido alejarnos de los numerosos datasets compuestos por vídeos sencillos y con numerosas restricciones.

La minimización de la energía formulada proporciona la solución de menor coste del problema, esto es, el etiquetado óptimo de los frames que forman el dataset. Para llevar a cabo la minimización se ha utilizado un algoritmo de corte de grafos, especialmente apropiado para funciones de energía no convexas de miles de dimensiones, como la nuestra.

Para evaluar el método creado se han utilizado los datasets *KTH* y *Hollywood2*. Del análisis de los resultados del proceso de parametrización y de la minimización de la energía resultante se puede concluir lo siguiente:

- Los resultados obtenidos en el dataset *KTH* son mejores, tanto en la segmentación como en el reconocimiento de acciones. Se debe principalmente al menor número de acciones contenidas en este dataset (6 acciones, frente a las 12 del *Hollywood2*) y a la menor complejidad de los vídeos, ya que las acciones tienen lugar en un menor número de escenarios diferentes y son llevadas a cabo por un menor número de sujetos distintos. No obstante, un 38.49 % de precisión en la segmentación y un 50 % de acierto en el reconocimiento de acciones son resultados bastante buenos si se tiene en cuenta que se ha utilizado únicamente un vídeo anotado por acción.

- La adición de los términos de la energía basados en detección de personas y en acciones centradas mejora la segmentación y el reconocimiento obtenidos al utilizar únicamente los términos de bolsa de palabras. Aunque en el dataset *KTH* no haya podido comprobarse la mejoría de los resultados al añadir esos términos, por ser sus vídeos demasiado simples, en el dataset *Hollywood2*, compuesto por vídeos más realistas, el término de detección de personas y el de acciones centradas han demostrado ser útiles a la hora de mejorar los resultados.

- En el dataset *Hollywood2* se ha obtenido un 31.19 % de precisión en la segmentación y un 33.33 % de acierto en el reconocimiento. Aunque a primera vista estos resultados puedan parecer bajos, se debe tener en cuenta que, si se asignara a cada frame una de las 13 posibles etiquetas de forma aleatoria, la probabilidad de acierto sería del $1/13 = 7.69\,\%$. Además, la mayor precisión obtenida en [18] a la hora de reconocer las acciones en este dataset es del 47.70 %. Si tenemos en cuenta que ese resultado fue obtenido utilizando cientos de vídeos anotados y el nuestro ha sido obtenido con un sólo vídeo anotado por acción, es posible afirmar que el reconocimiento débilmente supervisado es una línea de investigación muy prometedora.

4.2. Líneas futuras

En el presente proyecto ha quedado claro el potencial existente en la segmentación y el reconocimiento de acciones en vídeos. Son muchas las aplicaciones ya creadas, pero continuamente surgen nuevas ideas para mejorar las existentes o desarrollar otras nuevas. Es por eso que se debe intentar profundizar cada vez más en el problema y encontrar métodos menos supervisados y que obtengan tasas de acierto mayores.

En base a los resultados expuestos y a las conclusiones extraídas, se han determinado dos posibles líneas de trabajo futuras:

- **Preprocesado de las escenas**

  Para que los métodos de reconocimiento de las acciones sean fiables, deben ser probados en datasets realistas, en los que los vídeos contengan gran variedad de acciones, escenarios, personajes... Un claro ejemplo de ésto es el dataset *Hollywood2*. En él se ha visto que en muchas ocasiones las acciones que se quieren reconocer no tienen lugar en posiciones centradas la escena, ni son llevadas a cabo de la misma forma. Hay abrazos que ocurren en primer plano, en una sencilla habitación, mientras que otros suceden en posiciones alejadas del punto de vista del espectador, en medio de una multitud.

  Si fuéramos capaces, de alguna forma, de determinar las regiones de las escenas en que es más probable que estén llevándose a cabo las acciones, se podría centrar la extracción de descriptores espacio-temporales en ellas. De esta forma sería necesario procesar un volumen de datos mucho menos y la información extraída de él sería mucho más rica y descriptiva. En la figura 4.1 se puede ver esta idea de forma gráfica.

- **Segmentación espacial de las acciones**

  Una vez sentadas las bases de lo que supone la segmentación temporal de las acciones en los vídeos, es inevitable plantearse la posibilidad de llevar a cabo una segementación espacial. Si es posible reconocer la acción que se está llevando a cabo en una región acotada de un determinado vídeo, debería ser posible segmentar en esos frames a las personas que la están realizando.

  Como la acción ha sido reconocida previamente, se podrían utilizar descriptores que modelen los movimientos del cuerpo de las personas al llevar a cabo dicha acción. De esta forma seríamos capaces de encontrar las regiones de los frames en las que están teniendo lugar movimientos similares y segmentar así a las personas. Esta idea se recoge en la figura 4.2.

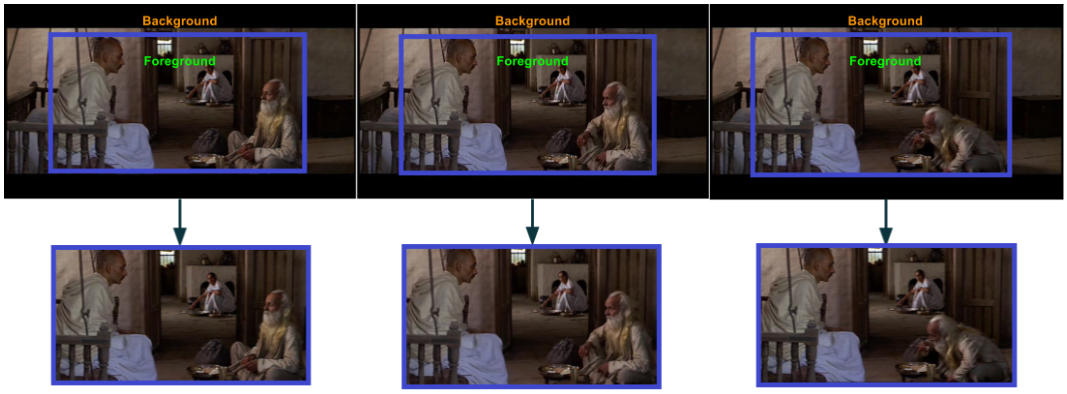

Figura 4.1: Preprocesado de las escenas. La idea es separar la región de la imagen que contiene información relevante (personas, movimiento, objetos...), o *foreground*, de aquella región carente de información de interés, o *background*.

Figura 4.2: Sementación espacial de las acciones. Si la acción ha sido previamente reconocida y segmentada temporalmente, debería ser posible segmentar a la persona o personas que la están llevando a cabo.

Bibliografía

[1] J. Sturm, E. Bylow, F. Kahl, and D. Cremers, "CopyMe3D: Scanning and printing persons in 3D," in *German Conference on Pattern Recognition (GCPR)*, Saarbrücken, Germany, September 2013.

[2] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[3] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3. IEEE, 2004, pp. 32–36.

[4] V. Delaitre, I. Laptev, and J. Sivic, "Recognizing human actions in still images: a study of bag-of-features and part-based representations." in *BMVC*, vol. 2, no. 5, 2010, p. 7.

[5] I. Laptev and P. Pérez, "Retrieving actions in movies," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on.* IEEE, 2007, pp. 1–8.

[6] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf, "Motion interchange patterns for action recognition in unconstrained videos," in *Computer Vision–ECCV 2012.* Springer, 2012, pp. 256–269.

[7] J. Zhu, B. Wang, X. Yang, W. Zhang, and Z. Tu, "Action recognition with actons," in *Computer Vision (ICCV), 2013 IEEE International Conference on.* IEEE, 2013, pp. 3559–3566.

[8] J. Guo, Z. Li, L.-F. Cheong, and S. Z. Zhou, "Video co-segmentation for meaningful action extraction," in *Computer Vision (ICCV), 2013 IEEE International Conference on.* IEEE, 2013, pp. 2232–2239.

[9] J. Luo, W. Wang, and H. Qi, "Group sparsity and geometry constrained dictionary learning for action recognition from depth maps," in *Computer Vision (ICCV), 2013 IEEE International Conference on.* IEEE, 2013, pp. 1809–1816.

[10] C. Nieuwenhuis and D. Cremers, "Spatially varying color distributions for interactive multi-label segmentation," vol. 35, no. 5, pp. 1234–1247, 2013.

[11] G. Kuschk and D. Cremers, "Fast and accurate large-scale stereo reconstruction using variational methods," in *ICCV Workshop on Big Data in 3D Computer Vision*, Sydney, Australia, December 2013.

[12] E. Toeppe, M. R. Oswald, D. Cremers, and C. Rother, "Image-based 3d modeling via cheeger sets," Queenstown, New Zealand, Nov. 2010, pp. 53–64.

[13] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ser. ICCV '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1150–. [Online]. Available: http://dl.acm.org/citation.cfm?id=850924.851523

[14] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[15] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," 2007.

[16] M. Marszałek, I. Laptev, and C. Schmid, "Actions in context," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.

[17] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the 15th international conference on Multimedia*. ACM, 2007, pp. 357–360.

[18] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, C. Schmid *et al.*, "Evaluation of local spatio-temporal features for action recognition," in *BMVC 2009-British Machine Vision Conference*, 2009.

[19] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–511.

[20] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[21] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, February 2004.

[22] Y. Boykov, O. Veksler, and R. Zabih, "Efficient approximate energy minimization via graph cuts," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1222–1239, November 2001.

[23] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *Proceedings of the International Conference on Computer Vision*, October 2009.

Índice de figuras

Índice de tablas

ANEXOS

Anexo A

Anotación de los vídeos

Se presenta aquí la anotación de los vídeos del dataset, llevada a cabo para poder evaluar los resultados de la minimización de la energía desarrollada. Se marcan con un asterisco (*) aquellos vídeos que forman parte del dataset reducido de 36 vídeos utilizado en la fase de prototipado de la energía.

#	**Vídeo**	**Acción**	$Frame_i$	$Frame_f$
1	actioncliptest00001	Kiss	95	114
2	actioncliptest00002	SitDown	15	55
3	actioncliptest00003*	HandShake	53	77
4	actioncliptest00004*	Eat	1	310
5	actioncliptest00008	SitDown	60	99
6	actioncliptest00009	SitUp	49	166
7	actioncliptest00010*	HandShake	55	84
8	actioncliptest00014	SitUp	12	81
9	actioncliptest00016	StandUp	40	77
10	actioncliptest00017*	HandShake	68	109
11	actioncliptest00018*	DriveCar	1	133
12	actioncliptest00020	Run	50	290
13	actioncliptest00023*	GetOutCar	16	119
14	actioncliptest00025*	Eat	1	239
15	actioncliptest00026	HughPerson	51	117
16	actioncliptest00027	HughPerson	48	133
17	actioncliptest00030	HughPerson	75	147
18	actioncliptest00031*	DriveCar	7	236
19	actioncliptest00033	SitUp	70	211
20	actioncliptest00035	Run	1	81
21	actioncliptest00039	StandUp	44	69

22	actioncliptest00044	Kiss	85	96
23	actioncliptest00045	SitDown	28	69
24	actioncliptest00048	Kiss	79	97
25	actioncliptest00051	SitUp	9	53
26	actioncliptest00052	SitDown	20	46
27	actioncliptest00053	Kiss	55	65
28	actioncliptest00054	Kiss	58	109
29	actioncliptest00055	Kiss	99	105
30	actioncliptest00062	Kiss	80	89
31	actioncliptest00063	SitUp	48	93
32	actioncliptest00064	Kiss	158	168
33	actioncliptest00065	Run	71	131
34	actioncliptest00067	SitDown	29	60
35	actioncliptest00072	Run	20	88
36	actioncliptest00073*	GetOutCar	19	74
37	actioncliptest00074*	HandShake	53	67
38	actioncliptest00075*	HandShake	54	159
39	actioncliptest00077	SitDown	29	126
40	actioncliptest00081*	FightPerson	38	264
41	actioncliptest00082	SitUp	15	42
42	actioncliptest00083*	AnswerPhone	16	66
43	actioncliptest00084	StandUp	22	40
44	actioncliptest00088	Run	21	151
45	actioncliptest00090*	DriveCar	9	153
46	actioncliptest00092	StandUp	16	49
47	actioncliptest00095	StandUp	35	59
48	actioncliptest00097*	GetOutCar	21	93
49	actioncliptest00098*	AnswerPhone	15	146
50	actioncliptest00102	SitDown	16	55
51	actioncliptest00107*	DriveCar	1	275
52	actioncliptest00108*	GetOutCar	21	121
53	actioncliptest00109	HughPerson	38	96
54	actioncliptest00110	HughPerson	49	356
55	actioncliptest00112	HughPerson	83	131
56	actioncliptest00113*	HandShake	55	72
57	actioncliptest00115	Kiss	52	56
58	actioncliptest00116	StandUp	42	66
59	actioncliptest00117*	GetOutCar	16	141

60	actioncliptest00125	Kiss	52	196
61	actioncliptest00127*	AnswerPhone	54	212
62	actioncliptest00130	HandShake	69	96
63	actioncliptest00131	SitDown	33	75
64	actioncliptest00132*	AnswerPhone	28	226
65	actioncliptest00135*	AnswerPhone	102	197
66	actioncliptest00136	SitUp	42	116
67	actioncliptest00138	StandUp	19	66
68	actioncliptest00140	Run	18	103
69	actioncliptest00142*	GetOutCar	31	114
70	actioncliptest00143*	AnswerPhone	38	164
71	actioncliptest00144*	Eat	16	320
72	actioncliptest00146*	DriveCar	5	148
73	actioncliptest00147	GetOutCar	18	95
74	actioncliptest00148	AnswerPhone	38	138
75	actioncliptest00149	HughPerson	50	194
76	actioncliptest00152*	FightPerson	36	191
77	actioncliptest00154*	DriveCar	1	154
78	actioncliptest00155	DriveCar	7	110
79	actioncliptest00158	DriveCar	9	166
80	actioncliptest00159*	Eat	11	254
81	actioncliptest00161	AnswerPhone	32	202
82	actioncliptest00166	Run	33	196
83	actioncliptest00168	DriveCar	1	200
84	actioncliptest00170	GetOutCar	27	135
85	actioncliptest00173	AnswerPhone	60	104
86	actioncliptest00174	AnswerPhone	66	200
87	actioncliptest00175	SitUp	124	201
88	actioncliptest00177	Kiss	40	52
89	actioncliptest00180	Run	33	350
90	actioncliptest00183	GetOutCar	19	90
91	actioncliptest00187	SitDown	60	105
92	actioncliptest00189	Run	1	126
93	actioncliptest00191	StandUp	37	59
94	actioncliptest00192*	Eat	7	287
95	actioncliptest00198	SitUp	13	83
96	actioncliptest00199	StandUp	30	67
97	actioncliptest00200	HughPerson	36	91

98	actioncliptest00201	Kiss	56	62
99	actioncliptest00205	DriveCar	1	233
100	actioncliptest00207	StandUp	29	50
101	actioncliptest00210	Run	16	100
102	actioncliptest00213*	Eat	5	314
103	actioncliptest00214	SitDown	55	135
104	actioncliptest00215	DriveCar	4	248
105	actioncliptest00217	Eat	210	258
106	actioncliptest00219	DriveCar	6	81
107	actioncliptest00230	SitDown	45	120
108	actioncliptest00233	GetOutCar	14	78
109	actioncliptest00234	GetOutCar	11	106
110	actioncliptest00236	HandShake	44	65
111	actioncliptest00237	StandUp	21	49
112	actioncliptest00239	AnswerPhone	51	199
113	actioncliptest00240	HandShake	129	156
114	actioncliptest00243	Run	1	47
115	actioncliptest00245	SitUp	37	66
116	actioncliptest00250	GetOutCar	43	121
117	actioncliptest00251*	FightPerson	60	102
118	actioncliptest00252	SitUp	63	85
119	actioncliptest00253	AnswerPhone	71	171
120	actioncliptest00255	StandUp	60	105
121	actioncliptest00261	Eat	1	121
122	actioncliptest00265	SitDown	17	55
123	actioncliptest00266	HandShake	55	89
124	actioncliptest00267	HandShake	58	87
125	actioncliptest00275	HughPerson	204	245
126	actioncliptest00280*	FightPerson	19	201
127	actioncliptest00286*	FightPerson	50	310
128	actioncliptest00294	SitUp	45	73
129	actioncliptest00314	HandShake	77	110
130	actioncliptest00324	HughPerson	63	235
131	actioncliptest00344	Run	1	10
132	actioncliptest00346	HughPerson	74	145
133	actioncliptest00348	HughPerson	70	262
134	actioncliptest00394*	FightPerson	88	124
135	actioncliptest00404	FightPerson	11	83

136	actioncliptest00423	FightPerson	17	241
137	actioncliptest00426	FightPerson	26	76
138	actioncliptest00456	Eat	16	149
139	actioncliptest00495	FightPerson	93	258
140	actioncliptest00507	FightPerson	24	58
141	actioncliptest00523	FightPerson	33	291
142	actioncliptest00625	Eat	8	61
143	actioncliptest00633	Eat	8	292
144	actioncliptest00672	Eat	1	189

Tabla A.1: Anotación de los vídeos que conforman nuestro dataset. $Frame_i$ y $Frame_f$ designan los frames iniciales y finales de la acción contenida en el vídeo, respectivamente. Los vídeos marcados con un asterisco forman parte también del dataset reducido.

Anexo B

Documentación del código

Se presenta a continuación la estructura del código desarrollado para este proyecto. Se ofrecen descripciones breves pero precisas de los principales archivos que lo conforman.

- **configuration.m**: script para la creación del archivo de configuración ***conf.mat***. Contiene todos los parámetros ajustables necesarios para seleccionar el experimento a realizar y la configuración del mismo.

- **SIFT 2D**
  - **Frames SIFT2D**: carpeta que contiene los frames utilizados para entrenar el vocabulario asociado a los descriptores espaciales.
  - **Descriptors_SIFT2D_BoW**: carpeta que contiene el vocabulario y los descriptores de bolsa de palabras basados en SIFT 2D de todos los frames de los vídeos del dataset.
  - **bhattacharyya.m**: función para el cálculo de la distancia de Bhattacharyya.
  - **normalize_energy_term.m**: función para normalizar los términos de la energía.
  - **create_frames_folder.m**: función para extraer los frames para el entrenamiento del vocabulario.
  - **sift2d.m**: función que se encarga de entrenar el vocabulario y extraer los descriptores de bolsa de palabras basados en descriptores SIFT 2D.
  - **create_pairwise_sift2D.m**: script para la creación del término binario de la energía basado en descriptores SIFT 2D.

- **create_unary_sift2D.m**: script para la creación del término unario de la energía basado en descriptores SIFT 2D.

- **SIFT 3D**
  - **Descriptors_Training_SIFT3D**: carpeta que contiene los descriptores SIFT 3D utilizados para entrenar el vocabulario asociado.
  - **Descriptors_SIFT3D**: carpeta que contiene los descriptores SIFT 3D de los frames de todos los vídeos del dataset.
  - **Descriptors_SIFT3D_BoW**: carpeta que contiene el vocabulario y los descriptores de bolsa de palabras basados en SIFT 3D de los frames de todos los vídeos del dataset.
  - **Position_Matrixes**: carpeta que contiene las matrices con las posiciones donde los descriptores SIFT 3D deben ser calculados.
  - **Descriptor_Locations**: carpeta que contiene las localizaciones donde los descriptores SIFT 3D han sido finalmente extraídos.
  - **sift3d.m**: función encargada de extraer los descriptores de bolsa de palabras basados en descriptores SIFT 3D.
  - **normalize_energy_term.m**
  - Funciones asociadas a la extracción de los descriptores SIFT 3D.
  - **calculate_position_matrixes.m**: script que determina las posiciones donde los descriptores SIFT 3D deben ser extraídos en cada frame.
  - **sift3d_vocabulary_training.m**: script encargado de extraer los descriptores y entrenar el vocabulario.
  - **create_pairwise_sift3D.m**: script para la creación del término binario de la energía basado en descriptores SIFT 3D.
  - **create_unary_sift3D.m**: script para la creación del término unario de la energía basado en descriptores SIFT 3D.

- **Faces & Body Detection**
  - **People_Detections**: carpeta que guarda las detecciones de personas en los vídeos.
  - **normalize_energy_term.m**

- **people_detector.m**: función encargada de las detecciones de personas en los frames de todos los vídeos del dataset.
- **faceVSbody.m**: script que compara los resultados obtenidos por del detector de caras y el detector de cuerpos.
- **create_unary_people.m**: script para la creación del término unario de la energía basado en la detección de personas.

- **Action Centered**
  - **normalize_energy_term.m**
  - **create_unary_ac.m**: script para la creación del término unario basado en las acciones centradas.

- **GCMinimization**
  - **Test_Videos_12x12**: carpeta que contiene los 144 vídeos del dataset creado.
  - **Test_Videos_6x6**: carpeta que contiene los 36 vídeos del dataset reducido.
  - **Annotations**: carpeta que contiene las anotaciones de los vídeos.
  - **Energy_Terms**: carpeta que almacena los términos de la energía de los diferentes experimentos.
  - **GroundTruth**: carpeta que contiene los *GroundTruth* calculados.
  - **Results**: carpeta que almacena los resultados de los diferentes experimentos.
  - **normalize_energy_term.m**
  - **create_segclass.m**: función que calcula el vector *segclass* itilizado en la minimización.
  - **create_GTruth.m**: función para la creación del *GroundTruth* en función del experimento llevado a cabo.
  - **evaluate_results.m**: función para la evaluación de los resultados obtenidos en la minimización.
  - **annotate_videos.m**: script para la anotación de los vídeos del dataset.

- **GC_PFC.m**: script para la minimización de la energía. Es el archivo principal del código y el que debe ser ejecutado tras cambiar los parámetros deseados en la configuración. Se encarga, dependiendo del experimento seleccionado, de cargar en memoria los términos de la energía pertinentes y llevar a cabo la minimización de la energía resultante mediante *Graph Cuts*.