Capítulo 5

Vídeo vigilancia automática

Es habitual la presencia de cámaras dentro y fuera de los edificios. Operadores humanos deben realizar una monitorización contínua para poder dar una señal de alarma cuando ocurre un acto fuera de la normalidad. Por ello, cada vez es más necesario el desarrollo de sistemas automáticos de vídeo vigilancia. Sin embargo, es un problema de gran dificultad por la diversidad de escenarios y condiciones que tienen los sistemas de vídeo vigilancia. Para abordar el problema completo, se divide básicamente en tres partes: localización, seguimiento visual y descripción. En este capítulo se repasan los métodos más utilizados de localización de objetos. Se muestra cómo el modelado de escenas es la metodología más extendida y se presenta un algoritmo que resuelve el problema de trabajar con sistemas con cámara activa. A continuación se utiliza el algoritmo *iTrack* para realizar la fase de seguimiento visual. Su adaptación al problema consistirá en la definición de una densidad *prior* a partir de los resultados del algoritmo de localización. Se propondrán un conjunto de medidas con las que evaluaremos el rendimiento del algoritmo en secuencias de vídeo vigilancia. Estas secuencias forman parte de un estándar que se está desarrollando para la evaluación de los algoritmos de seguimiento visual en aplicaciones de vídeo vigilancia. Finalmente, se propone un método de representación de actividades humanas. A partir de esta representación es posible realizar una descripción de alto nivel de lo que ocurre en la escena.

5.1 Sistemas de vídeo vigilancia.

Los sistemas de vídeo vigilancia se están extendiendo cada vez más en la sociedad moderna. Es común verlos instalados en centros comerciales, bancos, estaciones de metro y de tren, aeropuertos y edificios del gobierno. El objetivo de estos sistemas es la monitorización de las acciones de las personas y los vehículos que están dentro de una zona de interés.

Sin embargo, en la mayoría de casos sólo se utilizan como sistemas de almacenamiento de imágenes que sirven como herramienta forense después de que el hecho

haya ocurrido. Normalmente, es necesaria la presencia de un operador humano que monitorice todas las imágenes para poder actuar en tiempo real avisando a los oficiales de seguridad de que puede haber un acto anormal.

Es clara la necesidad de un sistema automático de monitorización ya que en muchas ocasiones no es posible controlar todas las cámaras del sistema de vídeo vigilancia porque no se dispone de suficientes recursos humanos. Además, un operador humano que esté observando un conjunto de cámaras de forma contínua, es habitual que se aburra rápidamente y pierda la concentración.

Desde el punto de vista técnico, un sistema automático de vídeo vigilancia incluye tareas de localización de objetos, seguimiento visual, clasificación de objetos y reconocimiento de actividades y acciones humanas. Estas tareas se pueden abordar utilizando técnicas de visión por computador, reconocimiento de patrones e inteligencia artificial. La localización implica la detección de los objetos en la escena de interés, que deben ser clasificados para conocer su tipo, por ejemplo, personas o coches. Una vez localizados, el módulo de seguimento visual se encarga de mantener sus trayectorias a lo largo del tiempo. Finalmente, los modulos de reconocimiento de actividades y de acciones se encargan de realizar una descripción simbólica de lo que está ocurriendo en la escena.

El interés por los sistemas de vídeo vigilancia automáticos se demuestra en el creciente número de proyectos de investigación desarrollados por diferentes gupos internacionales. Entre los más importantes podemos citar el VSAM[1] americano, y los europeos AVS-PV[2] y ADVISOR[3].

La mayor dificultad de una aplicación de vídeo vigilancia automática es la diversidad de escenarios y de condiciones que tienen los sistemas de adquisición utilizados. Podemos encontrar sistemas con una o varias cámaras, que pueden ser estáticas o móviles, y diferentes tipos de sensores, por ejemplo, cámaras color o infrarrojas.

En primer lugar repasaremos los métodos de localización de objetos y se propondrá un nuevo método para modelar escenas de sistemas con cámara activa. A continuación se mostrará la aplicación del algoritmo *iTrack*, definido en el capítulo anterior, para realizar el seguimiento visual. Veremos su utilidad en esta aplicación debido a que es posible tener diferentes escenarios, con lo que los métodos basados en contexto se tendrían que ajustar a cada uno de ellos. Además, utilizaremos medidas de rendimiento específicas para este tipo de aplicación que nos permitirán una mejor evaluación del funcionamiento del algoritmo. Finalmente, se propondrá una representación de actividades humanas que es posible utilizar para realizar su reconocimiento en tiempo real.

[1] http://www.cs.cmu.edu/vsam/.
[2] http://www-sop.inria.fr/orion/orion-eng.html.
[3] http://www-sop.inria.fr/orion/ADVISOR/.

5.2 Localización.

El primer paso que debe realizar un sistema de vídeo vigilancia automática es la localización de los objetos de interés. Existen dos aproximaciones básicas para realizar esta tarea: la detección de movimiento y el modelado de la escena. La primera se puede realizar por medio de diferencias de imágenes[53] o análisis del flujo óptico[17]. La diferencia de imágenes es muy adaptativa en ambientes dinámicos, pero sólo extrae las zonas de movimiento y no todo el objeto de interés, ver Fig. 5.1. Además no funciona de manera correcta cuando existe un movimiento de la cámara. En este último caso, utilizando técnicas de flujo óptico es posible detectar el movimiento global de la cámara para diferenciarlo del movimiento individual de cada objeto. Sin embargo, la mayoría de métodos de flujo óptico son costosos computacionalmente y son muy inestables por el problema de la apertura[4].

La segunda aproximación básica consiste en realizar un modelo de la escena estática[4]. Esta aproximación permite localizar todos los objetos que no están en el modelo, es decir, que han aparecido en la escena, ver Fig. 5.1. Normalmente se realiza un modelo estadístico de la escena a partir de un conjunto inicial de imágenes que no contiene ningún objeto de interés. En [34] se modela cada píxel con el mínimo, el máximo y la máxima diferencia entre frames consecutivos. En [90] cada píxel de la escena queda definido por la media y la varianza de su valor en los frames de aprendizaje. Una extensión de este método para escenas complejas se presenta en [81], donde se utiliza un modelo de *Mixtura de Gaussianas* para cada píxel. Además se presenta un algoritmo iterativo para adaptar el modelo de forma eficiente durante el tiempo de proceso del método de vídeo vigilancia. Por último, también se han utilizado modelos estadísticos más complejos, como el análisis de componentes principales[69], para crear el modelo de escena.

(a) Imagen original. (b) Diferencia de imágenes. (c) Modelo de escena.

Figura 5.1: Resultados de la aplicación de las dos aproximaciones básicas de localización de objetos.

Todos estos métodos de modelado de escenas, sirven para sistemas básados en cámara estática con parámetros no modificables. No son directamente aplicables a un sistema con cámara activa, ya que la escena visible por la cámara cambia cuando se modifican sus parámetros. Para adaptar los métodos anteriores se han utilizado

[4]Normalmente la escena estática se corresponde con el fondo, es por ello que estos métodos también se conocen como métodos de *Background Subtraction*[87].

diferentes alternativas. La más directa es crear una matriz de imágenes, una para cada posible posición de la cámara, pero es impracticable por la gran cantidad de parámetros de la cámara. En [94] se define el conjunto mínimo de estados de la cámara, de manera que sean suficientes para abarcar todo el área de vigilancia. De esta manera, sólo se guardan los modelos de escena para este conjunto mínimo de estados. Durante el seguimiento, los movimientos de la cámara están restringidos a este conjunto de estados. Una alternativa menos costosa es la construcción de una imagen panorámica[5] de toda la escena[88, 33, 62]. Para realizar la localización se escoge la zona del panorama según los parámetros que tenga la cámara en ese momento.

A continuación, repasaremos el modelo de *Mixtura de Gaussianas* y la construcción de panoramas. Finalmente, combinaremos ambas técnicas para crear un modelo de escena utilizando sistemas con cámara activa.

5.2.1 Algoritmo de Stauffer-Grimson.

El algoritmo de Stauffer y Grimson[81] es una extensión del método *Pfinder*[90], la idea principal es representar cada píxel de la escena utilizando un modelo de *Mixtura de Gaussianas* adaptativo. Se considera que los valores de intensidad o color de cada píxel a lo largo del tiempo, $\boldsymbol{I}_t$, forman un proceso estocástico, $(\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_t)$, que es posible modelar con una mixtura de K densidades Gaussianas. Es decir, que la densidad de probabilidad para $\boldsymbol{I}_t$ es:

$$p(\boldsymbol{I}_t) = \sum_{i=1}^{K} w_{i,t} \cdot p(\boldsymbol{I}_t|i) \ , \tag{5.1}$$

donde K es el número de Gaussianas, y $w_{i,t}$ es una estimación del peso de la i-ésima Gaussiana de la mixtura en tiempo t. Se puede interpretar $w_{i,t}$ como la probabilidad *prior* de que un dato haya sido generado por el componente i de la mixtura. Estos *priors* han de cumplir las siguientes condiciones:

$$\sum_{i=1}^{K} w_{i,t} = 1 \ , \tag{5.2}$$

$$0 \leq w_{i,t} \leq 1 \ . \tag{5.3}$$

Se asume que las densidades Gaussianas tienen matriz de covarianza circular, de forma que:

$$\Sigma_{i,t} = \sigma_{i,t}^2(\mathrm{Id}) \ ,$$

donde Id es la matriz identidad.

[5] Un panorama es una imagen con un gran campo de visión, y pueden generarse de diferentes maneras[84, 35].

Por tanto:

$$p(\boldsymbol{I}_t|i) = \frac{1}{(2\pi\sigma_{i,t}^2)^{\frac{n}{2}}} \exp\left\{-\frac{\|\boldsymbol{I}_t - \boldsymbol{\mu}_{i,t}\|^2}{2\sigma_{i,t}^2}\right\} \ , \tag{5.4}$$

donde $\boldsymbol{\mu}_{i,t}$ y $\sigma_{i,t}^2$ son la media y la varianza de la i-ésima Gaussiana en tiempo t; y n es la dimensión de $\boldsymbol{I}_t$.

Si el proceso estocástico puede ser considerado como estacionario, la forma estándar de encontrar los parámetros que modelan el proceso es el algoritmo EM (*expectation-maximization*) [6]. Sin embargo, si consideramos que es un proceso temporal, implementar el algoritmo EM para una ventana temporal de los datos es muy costoso. Por esta última razón, se utiliza el siguiente algoritmo adaptativo para encontrar los parámetros deseados.

Para tiempo t, se recoge el valor de la variable aleatoria, $\boldsymbol{I}_t$, y se comprueba a cual de las K densidades Gaussianas pertenece. El valor pertenecerá a una densidad si está dentro del intervalo de 3 veces su desviación estándar. Si no pertenece a ninguna de las K densidades, la densidad menos probable es reemplazada por una nueva densidad con media igual al valor recogido, varianza alta y un peso bajo.

Los pesos de las K densidades en tiempo t se adaptan según:

$$w_{i,t} = (1-\alpha)\, w_{i,t-1} + \alpha\, M_{i,t} \ , \tag{5.5}$$

donde α es el coeficiente de aprendizaje[6]. Otra manera de interpretar este parámetro, es que $1/\alpha$ define una contante de tiempo que determina un cambio significativo en el modelo. $M_{i,t}$ es 1 para la densidad encontrada y 0 para el resto. Después de esta aproximación, los pesos se normalizan ya que su suma ha de ser igual a 1.

El último paso es la actualización de la media y la varianza de la densidad encontrada, m; los parámetros del resto de densidades permanecen iguales. Las reglas de ajuste son las siguientes:

$$\boldsymbol{\mu}_{m,t} = (1-\rho)\, \boldsymbol{\mu}_{m,t-1} + \rho\, \boldsymbol{I}_t \ , \tag{5.6}$$

$$\sigma_{m,t}^2 = (1-\rho)\, \sigma_{m,t-1}^2 + \rho(\boldsymbol{I}_t - \boldsymbol{\mu}_{m,t})^T(\boldsymbol{I}_t - \boldsymbol{\mu}_{m,t}) \ , \tag{5.7}$$

donde

$$\rho = \alpha \cdot p(\boldsymbol{I}_t|m) \ , \tag{5.8}$$

es el factor de aprendizaje. Es posible ver estas reglas como la del ajuste de los pesos,

[6] Esta regla puede ser interpretada como $w_{i,t} = w_{i,t-1} + \alpha\,(M_{i,t} - w_{i,t-1})$.

Ec. (5.5), excepto que en este caso sólo se actualizan los parámetros de la densidad encontrada. El algoritmo completo se muestra en la Fig. 5.2.

Una de las ventajas de este algoritmo es que cuando un píxel entra a formar parte de la escena, no destruye el valor anterior del modelo. El valor de la escena original permanece en el modelo hasta que su peso disminuye y entra un nuevo valor en la escena. Así, si un objeto permanece estático el tiempo suficiente pasa a formar parte del modelo de escena. Pero cuando el objeto comienza a moverse de nuevo, el modelo recupera su valor original rápidamente.

Para estimar la escena más probable nos interesará el valor medio de la densidad Gaussiana con más peso y menos varianza. Por el algoritmo de establecimiento de parámetros definido, es lógico pensar que los píxels que permanecen estáticos aumentan su peso y disminuyen su valor de varianza. En contraste, cuando un nuevo objeto ocluye a la escena, en general, su valor no se corresponderá con ninguna de las densidades del modelo. Esto se traducirá en la creación de una nueva densidad que tendrá un valor alto de varianza mientras el objeto continue en movimiento.

Para tener en cuenta estos casos y decidir cual es el modelo de escena más probable, ordenaremos las densidades por el valor de w/σ. Este valor aumenta cuando la densidad es más probable y tiene menos varianza. Las primeras B densidades se escogen como modelo de escena, donde:

$$B = \arg\min_{b} \left(\sum_{k=1}^{b} w_{k,t} > T \right) \quad , \tag{5.9}$$

donde T es una medida de la mínima porción de los datos que ha de tenerse en cuenta para crear el modelo de escena. Si se escoge un valor pequeño de T, el modelo de escena suele ser unimodal. Para este caso, escoger directamente la densidad más probable ahorra tiempo de cálculo.

Los resultados obtenidos para diferentes secuencias para un modelo de $K = 2$ Gaussianas se muestran en la Fig. 5.3. Finalmente, para localizar los objetos de interés en la imagen, se escogen los píxels que no pertenecen al modelo de escena, ver Fig. 5.4.

Modelo de escena de Stauffer-Grimson

Sea $\boldsymbol{I}_t$:

1. Comprobar si el valor pertenece a una de las K densidades:

$$\|\boldsymbol{I}_t - \boldsymbol{\mu}_{i,t-1}\| < 3 \cdot \sigma_{i,t-1} \quad \forall i \in \{1, \ldots, K\} \ .$$

   (a) Si no pertenece a ninguna de las K densidades: reemplazar la densidad menos probable, densidad l con peso más bajo, por una nueva densidad con $\boldsymbol{\mu}_{l,t} = \boldsymbol{I}_t$, varianza inicial grande y peso bajo.

   (b) Si pertenece a una de las densidades: ajustar las K distribuciones como sigue:

$$w_{i,t} = (1 - \alpha)\, w_{i,t-1} + \alpha\, M_{i,t} \ ,$$

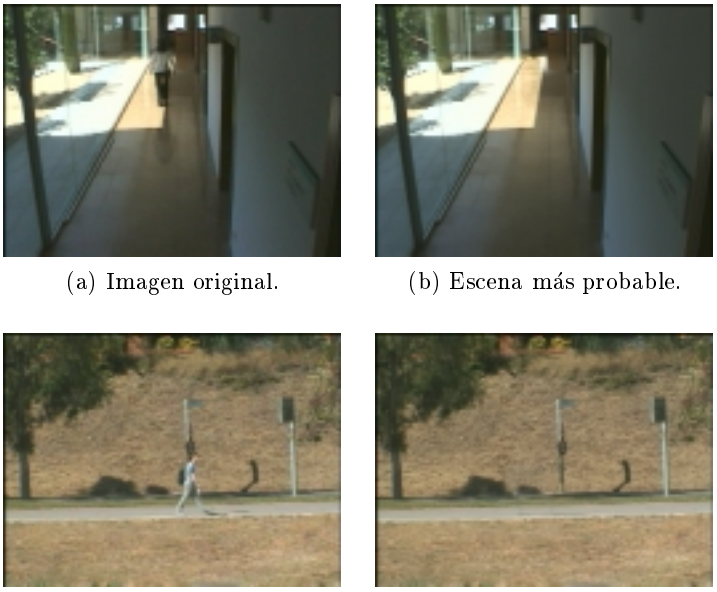
   donde $M_{i,t}$ es 1 para la densidad encontrada y 0 para el resto. Después normalizar todos los pesos. Actualizar media y varianza de la densidad encontrada, m, como sigue:

$$\boldsymbol{\mu}_{m,t} = (1 - \rho)\, \boldsymbol{\mu}_{m,t-1} + \rho\, \boldsymbol{I}_t \ ,$$

$$\sigma^2_{m,t} = (1 - \rho)\, \sigma^2_{m,t-1} + \rho (\boldsymbol{I}_t - \boldsymbol{\mu}_{m,t})^T (\boldsymbol{I}_t - \boldsymbol{\mu}_{m,t}) \ ,$$

   donde $\rho = \alpha \cdot p(\boldsymbol{I}_t | m)$, y α es el coeficiente de aprendizaje.

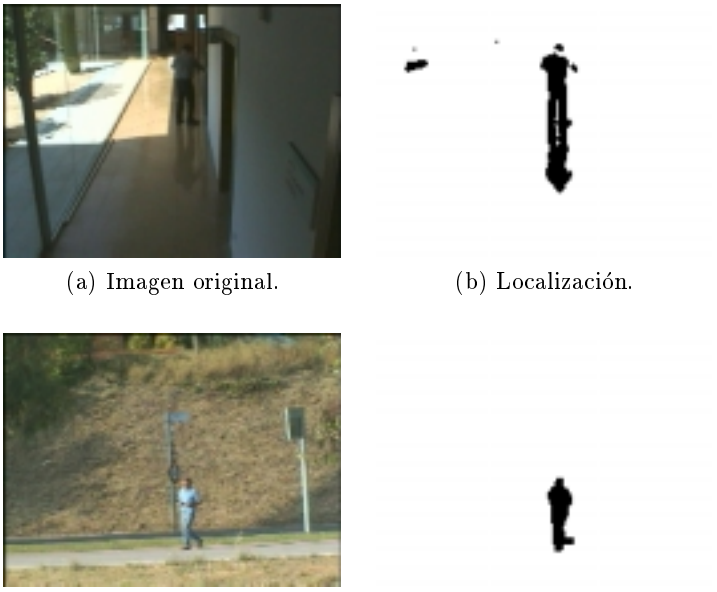
Figura 5.2: Algoritmo de ajuste del modelo de escena basado en *Mixtura de Gaussianas.*



(a) Imagen original. (b) Escena más probable.

(c) Imagen original. (d) Escena más probable.

Figura 5.3: Modelo de escena utilizando el algoritmo de Stauffer-Grimson para $K = 2$.



(a) Imagen original. (b) Localización.

(c) Imagen original. (d) Localización.

Figura 5.4: Resultados de la localización utilizando el algoritmo de Stauffer-Grimson para $K = 2$.

5.2.2 Creación de un panorama.

Los movimientos que puede hacer una cámara son finitos. Normalmente se pueden describir por medio de la composición de una rotación sobre un eje, una traslación y la variación de la distancia focal. De hecho, una cámara suele tener, como máximo, 7 grados de libertad:

- Posición cartesiana en el espacio: x, y, z.
- Orientación o dirección del punto de vista: *pan* (θ), *tilt* (ϕ) y *roll* (ψ).
- Distancia focal: f, relacionada con el *zoom*.

Una cámara de vídeo vigilancia suele ser estática, sin embargo, para cubrir el mayor área posible, tiene los movimientos de *pan* y *tilt*. Hemos limitado nuestro estudio a este tipo de cámaras[7]. Aprovecharemos que es posible conocer los parámetros de *pan* y *tilt* de la cámara para construir el panorama.

Si asumimos que la cámara rota alrededor de su centro óptico y que está situada en el centro de una esfera, la cámara observará el interior de la esfera. Desde el punto de vista de la cámara, cada punto de la escena se puede ver como un punto transformado geométricamente en la superficie interior de esta esfera.

Las transformaciones geométricas que realizaremos para obtener el panorama serán:

1. Tansformar cada imagen a la superficie interior de la esfera.
2. Convertir esta esfera en una superficie plana.

Para poder hacer estas transformaciones, se asume que se conocen el grado de *pan* que tiene la cámara, θ, el grado de *tilt*, ψ, y el campo de vista de la cámara en ambas direcciones, β y γ. Los grados de pan y tilt se pueden conocer a partir de la información que porporciona el *driver* de la cámara. Para saber el campo de vista, se realiza una calibración previa de la cámara[20]. Para proyectar cada punto (x, y) de la imagen en la superficie interior de la esfera hacemos la siguiente tranformación:

$$\theta_x = \theta + x \cdot \frac{\beta}{S_x} \ , \tag{5.10}$$

$$\psi_x = \psi + y \cdot \frac{\gamma}{S_y} \ , \tag{5.11}$$

donde S_x y S_y es el tamaño de la imagen horizontal y vertical respectivamente, y x e y son las coordenadas del punto a transformar escogiendo el $(0, 0)$ como el centro de la imagen, es decir, $x \in [\frac{-S_x}{2}, \frac{S_x}{2}]$ e $y \in [\frac{-S_y}{2}, \frac{S_y}{2}]$.

[7] Más concretamente hemos trabajado con el modelo EVI-D31 de Sony.

Para realizar la segunda transformación y convertir la esfera en una imagen plana, los ejes cartesianos de la imagen se corresponderán directamente con los valores de los ángulos de los puntos transformados. Un ejemplo del resultado de la creación de un panorama, utilizando 20 posiciones de pan y 5 de tilt, se muestra en la Fig. 5.5. El campo de vista en horizontal es de 100^o, y en vertical de 20^o. Para evitar las distorsiones que introduce la lente de la cámara se han recortado las imágenes tomando sólo la parte central de las mismas.

Figura 5.5: Panorama con un campo de vista de 100^o en horizontal y 20^o en vertical.

El panorama obtenido tiene dos problemas: hay cambios de iluminación en los diferentes fragmentos, y aparecen partes de objetos en movimiento. El primer problema es debido a que no utilizamos ninguna función de corrección para hacer uniforme la iluminación en todo el panorama. El segundo problema ocurre porque no se tienen en cuenta los objetos en movimiento que aparecen en la escena. En el siguiente apartado veremos como la construcción de un panorama a partir de un modelo de escena resuelve estos problemas.

5.2.3 *iLoc*: modelo activo de escena.

Nuestra aproximación para realizar el modelo de escena para cámara activa está basada en una combinación de las técnicas de creación de panoramas y de modelado de escenas. La escena se modelará como un panorama donde el valor de cada píxel se establece utilizando el algoritmo de Stauffer-Grimson. Para ello, cada imagen obtenida se registra en el panorama de la forma vista en el apartado anterior: proyectando la imagen al interior de una esfera y después transformando la esfera en un plano. Así, el panorama de la escena será la media de la densidad Gaussiana más probable para cada píxel.

Una vez modelada la escena, para realizar la localización de los objetos, se obtiene el fragmento del panorama que corresponde a los parámetros de *pan* y *tilt* con los que se ha adquirido la imagen. Para realizar este proceso, se indexa el panorama de manera inversa a como se ha creado. Para acelerar los cálculos se mantiene una LUT con la correspondencia entre la posición de cada píxel dentro de la imagen y su posición en el panorama para el *pan* y *tilt* actuales. En la Fig. 5.6 se muestra el

resultado de una escena modelada con esta técnica de indexación.

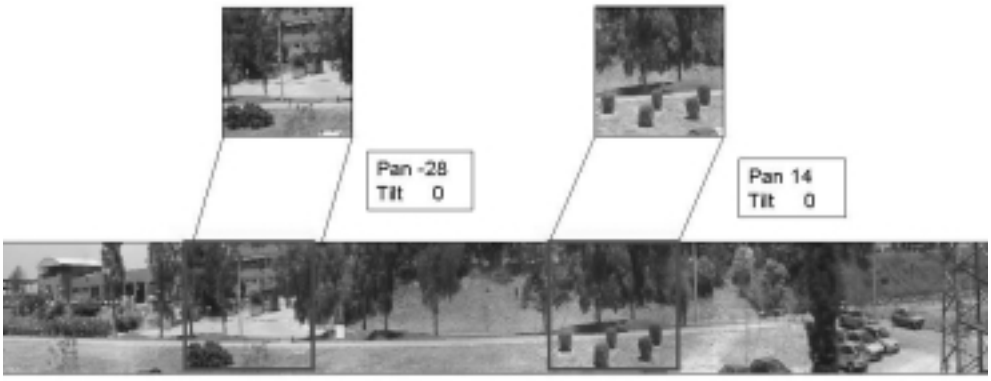


Figura 5.6: Esquema de indexación para la generación de un modelo de escena activo.

Los dos problemas que tiene la construcción de panoramas, la uniformidad de la iluminación y la aparición de objetos, quedan solucionados al utilizar el modelo de escena basado en *Mixtura de Gaussianas*. En cada fragmento de la escena que se registra es posible localizar los objetos en movimiento, estos se eliminan del proceso de construcción del panorama y no aparecerán en el modelo de escena final. Además, como el panorama de la escena se crea a partir de los valores medios de los fragmentos alineados, se elimina el problema del cambio de iluminación entre fragmentos consecutivos. Por estas dos razones, nuestro método proporciona una manera robusta de crear panoramas en la presencia de objetos en movimiento y cambios suaves de iluminación. Esto es una mejora respecto a los métodos de creción de panoramas que asumen una escena estática [84, 35]. Un ejemplo de panorama generado utilizando el modelo de escena se muestra en la Fig. 5.7.

5.2.4 Evaluación.

El método de localización basado en un modelo de escena activa, está implementado sobre una plataforma PC y puede procesar entre 14 y 17 frames por segundo sobre un Pentium a 800Mhz, con imágenes de una resolución de 192 $\times$ 144 píxels. Los resultados que aquí se presentan se han obtenido después de varias horas de funcionamiento del algoritmo en un entorno real, sin monitorización de un operador humano.

Para evaluar el sistema, se han definido dos clases de objetos a identificar: personas y no-personas. Este último tipo incluye coches y errores del sistema, como reflejos o pequeños movimientos de los objetos de la escena. Para realizar la clasificación nos hemos basado en una modelización del tamaño característico de la caja envolvente[8]

[8]Se ha modelado con una densidad Gaussiana bidimensional, donde una dimensión es la altura de la caja y otra la anchura.

Figura 5.7: Panorama generado utilizando el modelo activo de escena, *iLoc*. Campo de visión: 60^o en horizontal y 34^o en vertical. Resolución de cada fragmento: 192 × 144 píxels.

de los objetos localizados (*bounding-box*) y su continuidad temporal. Se han asociado por proximidad los objetos localizados durante k imágenes consecutivas. Se clasifican los objetos de forma individual durante las k imágenes y después se realiza un proceso de votación para los objetos asociados para decidir su tipo final.

Como medidas de evaluación del localizador nos interesa conocer el número de falsos positivos o índice de falsas alarmas, y el número de falsos negativos o índice de pérdidas. Sin embargo, si queremos conocer el índice de pérdidas es necesario monitorizar por parte de una persona todo el proceso de localización. Nos interesa definir un método de evaluación que no necesite una monitorización contínua. Para ello, nos centraremos en los falsos positivos. Si durante la fase de pruebas se almacenan todas las imágenes de los objetos localizados es posible determinar posteriormente el número total de objetos localizados y el número de falsas alarmas para cada tipo de objeto. Definiremos el índice de falsas alarmas, m_i, para cada tipo de objeto i como:

$$m_i = \left(1 - \frac{B_i}{A_i}\right) \quad , \tag{5.12}$$

donde A_i el número de objetos localizados de la categoria i, y B_i el número de objetos localizados correctamente. Cuanto más próximo a 0 sea m_i, mejor será la localización. El resultado de la evaluación después de 8 horas de funcionamiento del método se muestra en la Tabla 5.1.

	Obj localizados, A_i	Falsos positivos $(A_i - B_i)$	m_i
Personas	82	22	0.27
Otros	993	70	0.07

Tabla 5.1: Evaluación del Localizador.

Hay que resaltar que aunque hay errores, la mayoría no son de importancia. Hay que tener en cuenta que aunque una persona no sea clasificada correctamente en los primeros k frames, si puede serlo en los siguientes. En la Fig. 5.8 se pueden observar algunos de los objetos clasificados como personas.

Figura 5.8: Personas localizadas.

5.3 Seguimiento visual con *iTrack*.

El módulo encargado de mantener las trayectorias de los objetos localizados a lo largo del tiempo es el de seguimiento visual. La mayoría de sistemas de vídeo vigilancia automática utilizan el Filtro de Kalman. Sin embargo, debido a la unimodalidad de este filtro, para realizar el seguimento de múltiples objetos es necesaria la utilización de múltiples filtros y heurísticas ad-hoc para asociar las medidas obtenidas con cada uno de los filtros[34, 40, 53, 81, 19].

Por otro lado, existe la posibilidad de utilizar el Filtro Bayesiano, implementado con partículas, que permite el seguimiento de múltiples objetos. En [57] se presenta un sistema de seguimiento visual de múltiples objetos, utilizando como base el algoritmo CONDENSATION[42]. Esta aproximación está definida para el seguimiento visual de formas y está limitada a dos objetos. Para sistemas de vídeo vigilancia, se ha utilizado una aproximación jerárquica como variación del Filtraje Bayesiano básico, aplicado en un problema de seguimiento de personas[85]. Por último, recientemente se ha propuesto una extensión del algoritmo CONDENSATION para el tratamiento de múltiples objetos, donde el *prior* se construye a partir de un modelo de escena [45].

En todos estos trabajos previos, el denominador común es la dificultad del problema debido a la característica básica de los sistemas de vídeo vigilancia: la diversidad de escenarios. Como se ha mostrado en el capítulo anterior, uno de los objetivos básicos del algoritmo *iTrack* es tomar como observaciones directamente los valores de apariencia de la imagen. Esto permite su aplicación en diversos escenarios y con diferentes objetos, sin la necesidad de realizar un aprendizaje previo. Otro de los objetivos es su utilización para realizar el seguimiento de múltiples objetos sin un proceso previo de asociación de datos, con lo cuál es posible su utilización en esta aplicación.

La forma más directa de aplicar el algoritmo *iTrack* a las aplicaciones de vídeo vigilancia es por medio de la utilización del método de localización como densidad *prior* del algoritmo. En cada iteración del algoritmo, el método de localización actuará como *prior*, permitiendo la generación de nuevas muestras. Estas muestras pertenecerán a los objetos que estaban siendo seguidos o a nuevos objetos que aparecen en la escena. Por otro lado, la utilización del algoritmo *iTrack* en una aplicación de vídeo vigilancia, permite realizar una evaluación más completa del rendimiento del algoritmo.

5.3.1 Definición de la densidad *prior*.

En la definición del algoritmo *iTrack* se utilizaba una densidad de probabilidad *prior* para inicializar el método. Además, si es posible disponer de esta densidad durante todo el proceso de seguimiento, también actuará como función de reinicialización si ocurre un error grave en el proceso de estimación. Por ejemplo, en el caso de una larga oclusión del objeto de interés. Nuestro planteamiento consiste en la utilización del algoritmo de localización presentado en la sección anterior como densidad *prior*.

El algoritmo de localización cumple los dos requisitos que se exige para la función *prior*: permite la inicialización del seguimiento de los objetos y podemos disponer de ella en todas las imágenes de la secuencia. Antes de poder usarlo como densidad *prior* tenemos que expresar sus resultados de forma probabilística.

En primer lugar, recordaremos que el estado de un objeto en tiempo t está definido por $\boldsymbol{s}_t = (\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_t)$, donde $\boldsymbol{x}_t$ es el centro de la caja envolvente mínima que contiene al objeto de interés[9], $\boldsymbol{u}_t$ es la velocidad del objeto, y $\boldsymbol{w}_t$ es el tamaño de la caja.

Utilizando el algoritmo de localización es posible identificar qué píxels de la imagen pertenecen a los objetos de interés. Agrupando estos píxels se contruyen regiones conectadas denominadas *blobs* que se corresponderán con los objetos de interés de la escena.

Sea B el número de *blobs* detectado, el *prior* para la posición de los objetos de interés en la imagen se define como una *mixtura de Gaussianas*:

$$p_t(\boldsymbol{x}) = \sum_{k=1}^{B} P(k)p(\boldsymbol{x}|k) \ , \tag{5.13}$$

donde $P(k) = 1/B$, y $p(\boldsymbol{x}|k) = \eta(\boldsymbol{b}_k, \Sigma_B)$. $\boldsymbol{b}_k$ es la posición media de cada *blob* y Σ_B es común para todos los *blobs* y se establece a partir del error del método de localización en la escena de interés. $p_t(\boldsymbol{x})$ actuará como densidad *prior* para los componentes de posición del estado del objeto.

Con la imagen de *blobs*, también se define el *prior* para los componentes de tamaño de los objetos. Simplemente se define como una densidad Gaussiana de media igual al tamaño del blob y varianza dependiente del tipo de objeto. Por ejemplo, en el caso de personas, tendrá más varianza la anchura, debido al movimiento de la piernas al caminar.

Para generar la densidad *prior* para el componente de velocidad existen tres posibilidades:

1. Inicializar la velocidad a 0. El problema de esta suposición es que el objeto suele aparecer en la escena en movimiento, es decir, con una cierta velocidad inicial. En este caso, esto provocaría una diferencia muy grande en la primera estimación de la velocidad y el algoritmo tardaría en estabilizarse. La consecuencia es que el modelo de apariencia será muy ruidoso y el algoritmo no funcionaría correctamente.

2. Utilizar el flujo óptico. Es posible tener una estimación inicial de la velocidad del objeto usando un algoritmo de flujo óptico para estimar el desplazamiento medio que tienen los píxels pertenecientes al objeto. El problema de esta aproximación es que se tendría que realizar este cálculo en cada imagen de la secuencia y

[9] De hecho se puede interpretar como la posición del objeto dentro de la imagen.

ralentizaría todo el proceso. Además, en el caso de oclusiones del objeto de interés, esta estimación sería errónea.

3. Consistencia temporal. Generalmente, la primera aparición de un objeto en la escena no se considera directamente como un objeto a seguir. Una posibilidad es localizar este objeto en varias imágenes seguidas para intentar asegurar que no sea una región ruidosa. Por tanto, hasta que no ha sido localizado en k imágenes consecutivas el objeto se etiqueta como "Aparición" y el seguimiento consiste en una asociación simple. Si en algún momento el objeto no se puede asociar, se elimina de la lista de asociaciones y se considera que no era un objeto real. Durante este tiempo de aparición se puede hacer una estimación inicial de su velocidad.

Se han probado todas las posibilidades y al final se ha optado por la última opción. Hemos comprobado que para realizar un seguimiento visual correcto es importante la inicialización del objeto. La consistencia temporal asegura una mejor inicialización porque utilizamos más imágenes. El único problema es que el objeto tarda más en comenzar a ser seguido.

5.3.2 Evaluación.

Uno de los primeros trabajos que estudiaron la evaluación de los métodos de seguimiento visual en aplicaciones de vídeo vigilancia es el de Pingali y Segen [71]. Las medidas ideales para evaluar el rendimiento de un algoritmo de seguimiento visual requieren disponer de la trayectoria real de los objetos seguidos. En aplicaciones prácticas es muy difícil la obtención de las trayectorias reales de los objetos. Por esta razón, en este trabajo se proponen un conjunto de medidas prácticas, que es posible relacionar con la aplicación que se desea resolver.

Estas medidas prácticas se dividen en dos grupos:

- Medidas de cardinalidad.
- Medidas basadas en eventos.

Las medidas de cardinalidad se utilizan para comprobar si el número de objetos seguidos se corresponde con el número de objetos reales de la escena. Se definen dos medidas: el índice de pérdidas, m_f, y el índice de falsas alarmas, f_f:

$$m_f = \frac{\sum_t (a_t - r_t)}{\sum_t a_t}, \quad a_t > r_t \ , \tag{5.14}$$

$$f_f = \frac{\sum_t (r_t - a_t)}{\sum_t a_t}, \quad a_t < r_t \ , \tag{5.15}$$

donde a_t es el número real de objetos en el frame t, y r_t es el número de objetos proporcionado por el algoritmo. El problema de esta medida es que sólo se tiene en

cuenta el número de objetos. Es decir, no se realiza ninguna comprobación de que los objetos sean los mismos. Así, podría darse el caso de que en un instante de tiempo hubiera el mismo número de apariciones y desapariciones erróneas, lo que significaría un error grave del sistema, sin embargo, la medida daría un resultado correcto porque el número de objetos seguiría siendo el mismo.

Con las medidas basadas en eventos es posible comprobar la continuidad del seguimiento. Sin embargo, para no tener que anotar de forma manual toda la secuencia, se evalua la continuidad a partir de un conjunto de eventos. Estos eventos dependen del objetivo de la aplicación y de la escena de interés, pero en general tendrían que ser fáciles de obtener.

De forma general, un evento consiste en su etiqueta, e, y el instante de tiempo en el que ocurre, t. Los eventos observados serán los datos de referencia para realizar las medidas y se compararán con los eventos obtenidos por el sistema. Normalmente, coincidirán las etiquetas de los eventos, pero no el instante de tiempo en que ocurran.

En el trabajo de Pingali y Segen se proponen unas medidas basadas en la acumulación de ocurrencias de eventos de diferentes órdenes, donde el orden es el número de eventos seguidos. Estas medidas dan una buena aproximación del rendimiento del seguimiento en aplicaciones reales situadas en entornos donde aparecen gran cantidad de objetos. Principalmente, debido a que se basan en contar ocurrencias y que las secuencias contienen un número elevado de imágenes.

Actualmente, la evaluación de los algoritmos de seguimiento visual aún sigue siendo un problema abierto. Este hecho se demuestra por la reciente creación de un *Workshop* especializado para encontrar la respuesta a este problema[24]. Los dos temas principales a los que se dirige este *Workshop* son el desarrollo de un método de evaluación común y la creación de un conjunto de secuencias que permita realizar comparaciones entre diferentes trabajos.

De este conjunto, se ha seleccionado una secuencia de 20 segundos en la que aparecen 4 personas, ver Fig. 5.9. Utilizaremos esta secuencia para presentar los resultados del algoritmo *iTrack* utilizando como *prior* los resultados de la localización realizada a partir de un modelo de escena y para explicar el proceso de evaluación.

En este trabajo definiremos unas medidas basadas en eventos que nos sirvan para evaluar este tipo de secuencias más complejas, pero donde no aparecen gran cantidad de objetos. El tipo de medida propuesta es un intermedio entre las medidas reales de posición y las medidas basadas en eventos de Pingali y Segen. La idea es crear una tabla de eventos respecto al tiempo y compararla con la tabla de resultados proporcionada por el sistema.

Para definir las medidas se cuentan las correspondencias entre los eventos reales y los etiquetados por el sistema de visión en cada instante de tiempo. Los eventos que consideraremos son: “Aparición”, “Desaparición”, “Oclusión”, “Grupo”y “Rea-

Figura 5.9: Secuencia de test para la presentación de las medidas de evaluación.

parición". Estos eventos son los definidos en la sección 4.4 para la explicación de la extensión del algoritmo *iTrack* para múltiples objetos. Añadiremos un evento más para poder realizar la comparación: "En seguimiento". Este evento no se incluye en las tablas de eventos por motivos de espacio, en su lugar, se considera que entre los eventos de "Aparición" o "Reaparición"y cualquier otro evento el objeto está "En seguimiento".

Definiremos dos medidas, la primera nos permitirá comprobar el rendimiento del sistema para localizar y seguir objetos. La segunda medida es de precisión, buscaremos que además de que coincidan los eventos, las etiquetas de los objetos se correspondan. Así, se puede evaluar mejor la fase de seguimiento debido a que se comprueba el mantenimiento de la trayectoria de cada objeto. Estas medidas se definen como el tanto por ciento de imágenes para cada objeto donde existe una correspondencia correcta de eventos.

En el caso de la primera medida, C_n, un objeto que está en un determinado evento tiene correspondencia correcta cuando coincide con el evento encontrado por el sistema de visión. En la segunda medida, C_l, para una correspondencia correcta ha de coincidir además la etiqueta del objeto. Para verificar esta última medida, en la primera aparición de un objeto, éste se etiqueta como L_i, donde i es el número de objeto localizado por el sistema de visión. Si el sistema pierde este objeto por algún error del algoritmo, pero después lo vuelve a encontrar, cambiará su etiqueta inicial y no cumplirá el requisito de correspondencia de la segunda medida. Esta medida es útil para comprobar la continuidad del sistema de seguimiento. De todas maneras, un proceso posterior al seguimiento podría establecer correspondencias entre objetos con diferentes etiquetas y la primera medida sería suficiente.

Finalmente, mostraremos los resultados del algoritmo *iTrack* en comparación con el método W^4 de Haritaoglu et al.[31]. Se ha escogido este método porque también está basado en un modelo de escena y utiliza un método de estimación para mantener la trayectoria de los objetos. Sin embargo, W^4 se basa en un proceso heurístico previo de asociación de datos para encontrar la correspondencia entre cada objeto y el filtro al que pertenece. El método de estimación utilizado es el Filtro de Kalman. Para la secuencia de test, ver Fig.5.9, los eventos reales y los encontrados por ambos algoritmos se muestran en la Tabla 5.2.

t	Evento	W^4	*iTrack*
105	Aparición, P_1	Aparición, $L_1 = P_1$	Aparición, $L_1 = P_1$
115	Aparición, P_2	Aparición, $L_2 = P_2$	Aparición, $L_2 = P_2$
188	Aparición, P_3	Aparición, $L_3 = P_3$	Aparición, $L_3 = P_3$
208		Desaparición, L_2	
239			Desaparición, L_2
244		$L_3 = P_2$	
244		Aparición, $L_4 = P_3$	
279		Desaparición, L_1	
296		Aparición, $L_5 = P_1$	
322	Aparición, P_4	Aparición, $L_6 = P_4$	Aparición, $L_4 = P_4$
410	Desaparición, P_1	Desaparición, L_5	Desaparición, L_1
416		Grupo, (L_3,L_4)	
445	Desaparición, P_2	Desaparición, (L_3, L_4)	
450	Desaparición, P_3		Desaparición, L_3

Tabla 5.2: Comparación de resultados de la primera secuencia de test.

Para interpretar estos resultados de forma numérica, se realizan las medidas definidas, C_n y C_l, sobre los eventos de la Tabla 5.2. A continuación se detalla la forma de realizar las medidas sobre la tabla de eventos para el objeto P_1. Aplicando el algoritmo W^4 existe una correspondencia correcta para ambas medidas desde

$t = 105$ hasta $t = 279$, donde el algoritmo pierde al objeto. Éste vuelve a aparecer en el instante $t = 296$ y se mantiene hasta $t = 410$ con una etiqueta diferente. Por tanto, $C_n = 174 + 114 = 288$ y $C_l = 174$. Los resultados para todos los objetos y de forma global para la escena se muestran en la Tabla 5.3.

		W^4		*iTrack*	
Objeto	Imágenes	C_n	C_l	C_n	C_l
P_1	305	288	174	305	305
P_2	330	265	093	124	124
P_3	277	228	056	277	277
P_4	178	178	178	178	178
Total	1090	959	501	884	884
	100%	88%	46%	81%	81%

Tabla 5.3: Resultados correspondientes a los eventos de la Tabla 5.2.

En primer lugar se ha de destacar que la secuencia de test es muy ruidosa y compleja debido a la coincidencia de parte de la apariencia de las personas que intervienen en ella y la apariencia de la escena. Tal y como muestran los resultados de la Tabla 5.3, W^4 es un método que prioriza la fase de localización sobre la de seguimiento visual. Obtiene resultados correctos en la primera medida, pero en la medida de precisión disminuye de forma notable su rendimiento debido a los errores de la fase de localización. Estos errores del método de localización son debidos a las sombras de los objetos, ver Fig. 5.10, y a la vista perspectiva de la escena que provoca una oclusión entre dos objetos en la imagen que no ocurre en la escena real, ver Fig. 5.11.

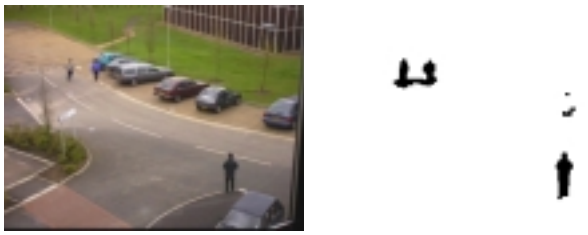


Figura 5.10: Error del método de localización debido a las sombras.

Sin embargo, los resultados de la medida de precisión, C_l, del algoritmo *iTrack* son mejores debido a que el método de estimación es más robusto que el Filtro de Kalman y las heurísticas de asociación de datos del método W^4. La desventaja en este caso es que la condición de consistencia temporal que utiliza el *prior* de *iTrack* es más rígida y provoca que un *blob* muy ruidoso no vuelva a aparecer.

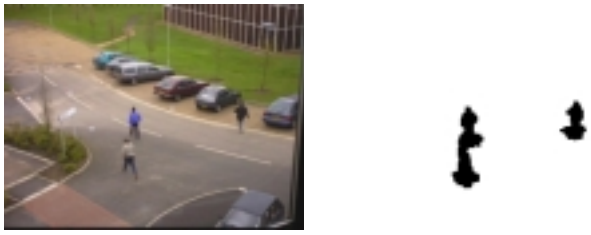


Figura 5.11: Error del método de localización debido a la perspectiva de la escena.

Como conclusión final de las pruebas realizadas comentar que una característica que hemos encontrado importante en la aplicación del algoritmo *iTrack* en este tipo de aplicaciones es que es muy sensible a su inicialización. Este hecho es debido a que la apariencia del objeto es muy importante puesto que es la característica básica del algoritmo, si el modelo de apariencia no se inicializa correctamente el objeto se pierde fácilmente.

5.4 *Keyframes*: reconocimiento de actividades.

Para completar el sistema de vídeo vigilancia automática debería realizarse una descripción semántica de la escena. Básicamente, existen dos posibilidades de hacer esta anotación, analizando las trayectorias temporales de los objetos o utilizando un método de reconocimiento de actividades.

El análisis de las trayectorias temporales es una forma de aprovechar de forma directa el resultado del módulo de seguimiento visual. Estos métodos se basan en la utilización de información adicional del área de vigilancia y de los objetos que pueden estar en ella. Después, combinan esta información previa con el resultado proporcionado por los módulos de visión por medio de técnicas de inteligencia artificial como redes Bayesianas[75, 2] o modelos ocultos de Markov[73, 15].

Existen dos aproximaciones básicas para realizar el reconocimiento de actividades humanas: las basadas en las características físicas y las basadas en la apariencia[1, 26, 13]. La primera aproximación implica la localización y el seguimiento de los miembros y las articulaciones de la persona. Se usa un modelo del cuerpo humano para mejorar el seguimiento visual al imponer las restricciones físicas al estado del objeto. Para ello, se requiere de un entorno controlado debido a la dificultad de identificar las partes del cuerpo humano en imágenes de vídeo real. De igual forma, es difícil que estos métodos funcionen en tiempo real.

La segunda aproximación se centra en usar modelos basados en apariencia para el movimiento humano. Las principales desventajas de estos modelos son la dependen-

Figura 5.12: Imagen extraída de la secuencia 23 (*Man running*) del libro *The Human Figure in Motion* de Eadweard Muybridge[61].

cia al punto de vista, la variabilidad de la apariencia (cambios en la ropa, sombras, tamaño del cuerpo y proporciones entre individuos), y el reconocimiento en presencia de oclusiones. Estas dificultades implican un dominio restringido de aplicación, es decir, escenas estáticas y donde las personas aparezcan siempre bajo el mismo punto de vista.

Siguiendo la idea básica de las aproximaciones basadas en la apariencia, a continuación se propone un método de descripción de actividades humanas. La idea es describir una actividad utilizando el mínimo conjunto de estados posible, donde cada estado se corresponde a una configuración determinada de la apariencia de la persona. Este conjunto de estados mínimo lo denominaremos *Keyframes*. Para mostrar de forma sencilla este concepto, considerar la imagen que se muestra en la Fig. 5.12. A partir de esta imagen es posible interpretar que la persona está corriendo. Este ejemplo es una forma simple de mostrar la posibilidad de reconocer actividades mediante la selección de unas pocas imágenes de toda la secuencia.

A continuación veremos una forma de realizar una descripción del cuerpo humano que soluciona el problema de los posibles cambios de apariencia. Después se muestra cómo obtener los *keyframes* de forma automática. Mediante la detección de estos keyframes en una secuencia, es posible saber en tiempo real qué actividad se está desarrollando. Para la descripción de esta última parte ver [27].

5.4.1 Descripción del cuerpo humano.

Antes de extraer los *keyframes* de la secuencia de una actividad, se requiere un preprocesamiento previo. En primer lugar, a partir de los módulos de localización y seguimiento visual es posible obtener la región de la imagen que contiene a la persona. Existen trabajos previos donde esta descripción basada en el concepto de *blob* es suficiente para conseguir el reconocimiento de actividades[90, 31, 14]. De todas formas, se basan en modelos 2D y en estadísticos de primer y segundo orden de los *blobs*.

Estas características no son las adecuadas para extraer las imágenes más repre-

sentativas de una actividad. Kovàcs sugiere una representación concisa de la forma basada en esqueletos [50]. En su trabajo, calculan una función basada en una métrica equidistante, básicamente, un mapa de distancias. De esta forma, calculan un esqueleto no uniforme con niveles de gris, donde los picos representan los puntos más lejanos al contorno. Con base en este trabajo, aplicamos este mapa de transformación al blob para obtener la representación final del cuerpo. En esta representación se reflejan claramente los cambios en la forma del cuerpo al realizar diferentes movimientos. Esta es una característica muy útil debido a que buscamos los cambios en la forma para seleccionar los más distintivos.

Figura 5.13: Segmentación y representación basada en el esqueleto de las personas.

Para nuestro objetivo, esta representación tiene otra ventaja: es independiente de la apariencia. Es decir, no importa si, por ejemplo, los actores llevan diferente vestuario, ya que no tendremos en cuenta los valores de apariencia de los píxels. Aplicar el mapa de distancias implica que sólo la forma de la persona será relevante para realizar la selección de *keyframes*. En la Fig. 5.13 se muestra un ejemplo de toda la fase de preprocesamiento.

5.4.2 El método de selección de *keyframes*.

Normalmente, en la secuencia de una actividad los movimientos más representativos corresponden a los menos probables. Esto significa que en la realización de una actividad la mayoría de movimientos se repiten. Los movimientos menos repetidos corresponden a movimientos extremos y son los que distinguen actividades diferentes. Nuestro objetivo es encontrar la descripción de estos movimientos.

Consideraremos que una actividad, A está formada por una secuencia de muestras $A = \{\mathbf{s}_1^A, ..., \mathbf{s}_N^A\}$, donde $\mathbf{s}_i^A$ es un vector formado por los valores de los píxels de la imagen del esqueleto ordenados por filas.

El primer paso para poder encontrar una descripción de la actividad, es encontrar el subespacio óptimo[10] para representar todas las posibles configuraciones que puede tomar esqueleto en algún momento de la realización de la actividad. Para ello, en primer lugar restaremos la media, $\bar{\mathbf{s}}^A$, a cada muestra:

$$\tilde{A} = \{\mathbf{s}_1^A - \bar{\mathbf{s}}^A, ..., \mathbf{s}_N^A - \bar{\mathbf{s}}^A\} \ . \tag{5.16}$$

[10] Utilizando como medida el error cuadrático.

Y calculamos la matriz de covarianza como:

$$Q = \tilde{A}\tilde{A}^T \ . \tag{5.17}$$

Los valores propios, λ_i, y los vectores propios, $\mathbf{e}_i$, de Q se calculan resolviendo la siguiente expresión[60]:

$$\lambda_i \mathbf{e}_i = Q\mathbf{e}_i \ . \tag{5.18}$$

Proyectando cada muestra, $\mathbf{s}_j^A$, en el subespacio representado por los vectores propios calculados, se obtiene la reprentación de la configuración del esqueleto en el espacio de la actividad:

$$\mathbf{y}_j^A = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_M]^T(\mathbf{s}_j^A - \bar{\mathbf{s}}^A) \ , \tag{5.19}$$

donde M es el número de vectores propios escogidos (normalmente $N < M$).

Dentro de este subespacio se define una distancia[59], que es un estadístico suficiente para caracterizar la probabilidad de que una muestra haya sido generada por esta actividad:

$$\hat{d}(\mathbf{s}_j^A) = \sum_{i=1}^{M} \frac{{y_{ij}^A}^2}{\lambda_i} + \frac{\epsilon^2(\mathbf{s}_j^A)}{\rho} \ , \tag{5.20}$$

donde y_i es la proyección de la muestra al subespacio de la actividad, λ_i son los valores propios, y M es la dimensión del subespacio. El término de la derecha es una estimación de la distancia al subespacio (*distance from feature space* -DFFS-), y se calcula a partir de la siguiente expresión:

$$\epsilon^2(\mathbf{s}_j^A) = \sum_{i=M+1}^{N} {y_{ij}^A}^2 = \| \mathbf{s}_j^A - \bar{\mathbf{s}}^A \|^2 - \sum_{i=1}^{M} {y_{ij}^A}^2 \ , \tag{5.21}$$

donde ρ es un peso que se encuentra minimizando la función de error:

$$\rho = \frac{1}{N-M} \sum_{i=M+1}^{N} \lambda_i \ . \tag{5.22}$$

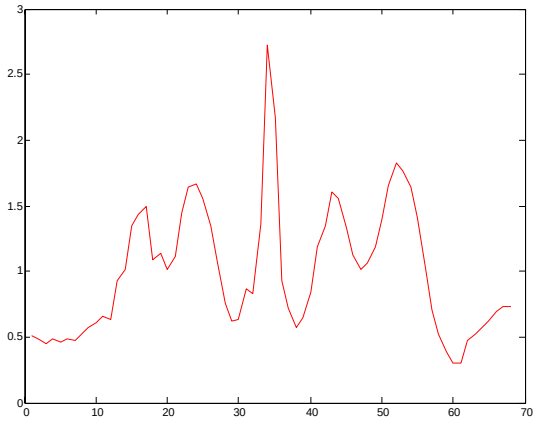


Figura 5.14: Medida de distancia en orden temporal.

Aplicando el orden temporal de las muestras de la actividad, es decir, tal y como se generan las muestras, obtenemos un gráfico de la distancia respecto al tiempo que utilizamos como función, ver Fig. 5.14. Los máximos locales de esta función corresponden a las muestras menos probables, es decir las configuraciones del esqueleto que ocurren menos veces en la secuencia de la actividad. Estos máximos constituyen la secuencia de *keyframes*, ver Fig. 5.15, también ordenados de forma temporal:

$$\mathbf{K}^A = \{\mathbf{k}_1^A, \mathbf{k}_2^A, ..., \mathbf{k}_L^A\} \ , \tag{5.23}$$

donde $\mathbf{K}^A$ es el conjunto de *keyframes* para la actividad A, y $\mathbf{k}_i^A$ es el i-ésimo em keyframe en orden temporal.

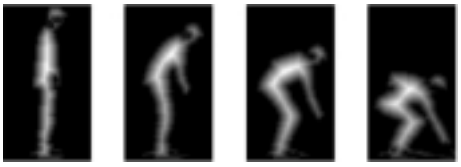


Figura 5.15: Conjunto de *keyframes* generado para la actividad de "agacharse".

La interpretación del método se basa en que los movimientos extremos se corresponden a los frames más representativos de una actividad. El resto de movimientos son repetitivos y se producen entre movimientos extremos. Desde un punto de vista probabilístico, los movimientos extremos son los menos probables de una secuencia. Este hecho es simplemente consecuencia del número de imágenes de la actividad

donde aparece dicho movimiento, menor que el número de movimientos repetitivos. De hecho, los movimientos menos probables son los que nos aportan más información para decribir una actividad.