

Trabajo Fin de Máster: Seguimiento de objetos en entornos concurrentes

Máster en Ingeniería Informática y de
Telecomunicación

Carlos Jesús Ruiz Fernández

Tutor: Miguel Ángel García García
Video Processing and Understanding Lab

Escuela
Politécnica
Superior

Universidad
Autónoma de
Madrid

2013

Contenidos

1. Introducción	3
2. Descripción del algoritmo.....	5
Modelo espacio-temporal	9
Predicción.....	14
Seguimiento.....	16
3. Detalles de la implementación	23
4. Resultados experimentales.....	25
5. Conclusiones	33
Ventajas.....	34
Inconvenientes	35
Mejoras.....	36
6. Bibliografía.....	37

1. Introducción

El seguimiento de objetos en secuencias de vídeo es actualmente un problema fundamental para áreas de aplicación tan importantes como la monitorización y la vídeo-seguridad, la Biometría, la Robótica o la interacción hombre-máquina, por citar algunas de las más destacadas. Es por ello que el estudio de todos los aspectos relacionados con este problema ha ido en aumento durante los últimos años, abriendo un amplio campo de posibilidades de investigación en nuevas técnicas de seguimiento de objetos y de mejora de los sistemas actuales.

Entre los problemas más destacados aún por resolver dentro de este ámbito se encuentra el seguimiento de objetos en entornos concurridos. La elevada complejidad de dichos entornos produce una degradación muy significativa de la eficacia de los algoritmos de seguimiento de propósito general. La investigación en este sentido se ha intensificado para obtener algoritmos cada vez más robustos y eficaces. Los entornos muy concurridos de personas presentan un alto riesgo de actividades peligrosas, entre las que cabe destacar el pánico, estampidas y accidentes que envuelven a un gran número de individuos y que están presentes muchas veces en los medios de comunicación. Para este tipo de situaciones, los algoritmos de seguimiento de objetos en secuencias de vídeo pueden ser un gran apoyo para el personal de seguridad, médico o de emergencias. Además, pueden ser utilizados en sectores tan dispares como el análisis de mercados, por ejemplo en centros comerciales, o en el análisis de rutas en aeropuertos o zonas comerciales.

Los vídeos de entornos concurridos presentan importantes retos todavía pendientes de resolver, entre ellos el tratamiento de un número tan elevado de objetos en movimiento y la frecuente oclusión parcial a la que estos están sometidos.

Uno de los algoritmos más recientes (1) ha demostrado tener una gran eficacia en este tipo de entornos. Se trata de una evolución de algoritmos anteriores, (2) y (3), que se han ido perfeccionando.

Algunos algoritmos de seguimiento asumen un fondo estático que permite diferenciar claramente el fondo y los objetos en movimiento. Esta restricción puede ser aceptable en entornos muy controlados y con pocos objetos, pero no es la situación habitual en el caso de escenarios muy concurridos. En este tipo de escenas y debido al gran número de personas presentes, el fondo suele ser muy reducido o bien directamente no ser visible.

Los vídeos de entornos muy concurridos suelen tener tres características que los definen: en primer lugar, el alto número de personas, que suelen dar

lugar a oclusiones parciales muy frecuentes. En segundo lugar, los individuos pueden moverse en un amplio número de direcciones dependiendo de sus propios objetivos, de sus vecinos más próximos y de los obstáculos físicos de la escena. Por último, la apariencia puede cambiar de forma significativa debido a los movimientos de los propios individuos.

La parte positiva de este tipo de entornos es que los movimientos de un objeto contribuyen al movimiento del conjunto de objetos de la escena. Se puede decir que las distintas trayectorias individuales tienden a repetirse a lo largo del tiempo en ciertas partes de la escena y en diferentes objetos. Por ejemplo, en la entrada a una estación de metro (**Fig. 1**), las trayectorias pueden variar pero habitualmente tendrán unas direcciones definidas claramente en cada parte de la escena. Estos movimientos terminan por definir una serie de patrones locales espacio-temporales.



Fig. 1. Dos escenas contiguas en el tiempo de la entrada a una estación de metro donde se pueden observar las distintas trayectorias de cada individuo. Como se puede observar, las trayectorias mayoritarias pueden definir un patrón claro en ciertas partes de la escena.

Es en este punto donde podemos definir el cambio de paradigma, desde uno centrado en el objeto, a un paradigma centrado en la escena. Esta visión permite definir el movimiento de los objetos de la escena en forma de patrones.

Una vez definidos estos patrones comunes, pueden ser aprendidos y servir posteriormente para hacer predicciones de movimiento individuales en regiones concretas de la escena.

En este trabajo se ha realizado el análisis, implementación y evaluación del algoritmo propuesto por L. Kratz y K. Nishino (1), que es uno de los más relevantes en la literatura reciente sobre seguimiento en entornos densos.

2. Descripción del algoritmo

El algoritmo de seguimiento de objetos en entornos concurridos que se ha analizado en este trabajo realiza el modelado de la estructura espacial y temporal del movimiento de los objetos de la escena.

La información del vídeo que se utiliza es únicamente la intensidad de cada píxel. Incluso si el vídeo dispone de los tres canales de color RGB, el análisis se realiza convirtiendo la información cromática a niveles de gris y procesando solamente dicho canal acromático.

A grandes rasgos, el algoritmo analizado define un marco bayesiano de seguimiento en el que primero se modela el movimiento de la muchedumbre mediante un modelo estadístico de los patrones de movimiento espacio-temporales, después se usa ese modelo para predecir los patrones espacio-temporales del objeto que se está siguiendo y finalmente se usa dicha predicción para mejorar el rendimiento de un algoritmo de seguimiento basado en filtro de partículas.

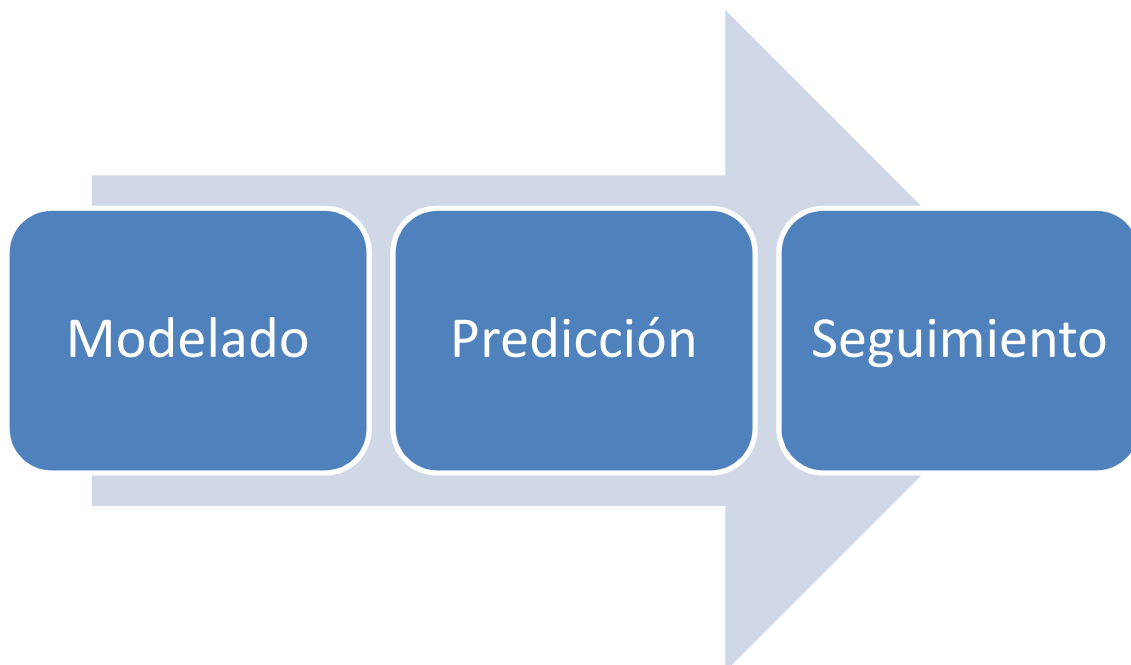


Fig. 2. Etapas del algoritmo analizado, que comienza con el modelado del movimiento de la muchedumbre, para después predecir una serie de patrones que finalmente serán los parámetros de entrada de un algoritmo de seguimiento basado en filtro de partículas.

Un patrón de movimiento viene definido por una serie de datos que definen el tipo de movimiento dentro de un modelo estadístico definido para la escena.

La escena completa puede tener muchos patrones diferentes de movimiento. Por eso es interesante destacar distintas zonas de actividad dentro de las imágenes para después definir los distintos patrones que se aplican a cada zona por separado. En este tipo de escenas, los patrones en determinadas áreas suelen ser bastante estables, representando flujos de movimiento bien definidos. De ese modo, es factible realizar una predicción posterior, facilitando así en gran medida el seguimiento de objetos en dichas áreas. Otras zonas pueden representar un reto mayor a la hora de definir patrones fiables para las mismas. Por eso la elección de estas zonas de actividad es una tarea relevante.

Para la selección de las zonas de actividad se define una cuadrícula regular (**Fig.3**). Esta cuadrícula delimita una serie de regiones rectangulares que particionan la escena de forma equilibrada. Para introducir la variable temporal en cada una de estas zonas es necesario considerar secuencias de imágenes capturadas en instantes de tiempo consecutivos. Esto permite la definición del concepto de *cuboide*. Un cuboide es un subvolumen del vídeo con dimensión espacial y temporal. En concreto, un cuboide es una matriz tridimensional de píxeles que viene definida por una posición en la imagen, un alto, un ancho y un número consecutivo de imágenes (cuadros), que representa un intervalo de tiempo.

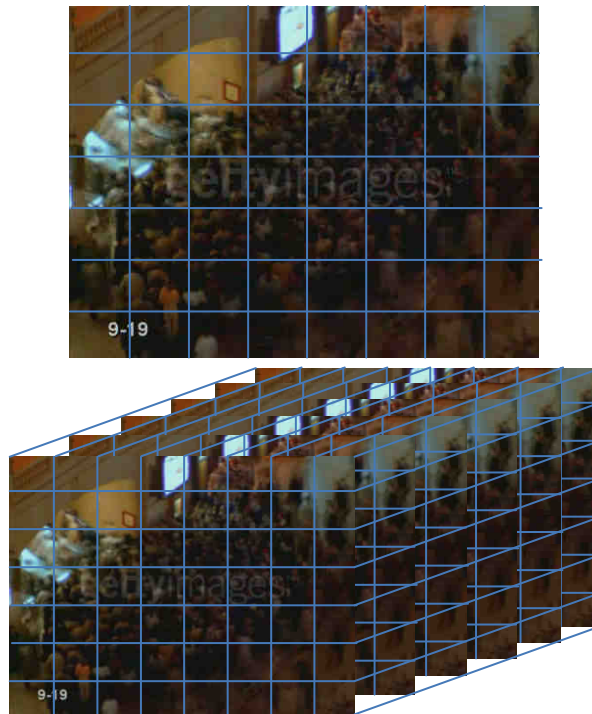


Fig. 3. Definición de la cuadrícula y posterior ampliación temporal para obtener los cuboides. Un cuboide está definido por un alto, un ancho, una posición en la imagen y un número de imágenes (cuadros) consecutivos, que definen una ventana temporal.

Una vez se ha realizado la segmentación de la secuencia de vídeo en cuboides, para cada una de las zonas de la cuadrícula se realiza el entrenamiento considerando cada secuencia de cuboides ordenados en el tiempo (**Fig. 4**). A dichas secuencias de cuboides correspondientes a una misma zona de la imagen se le denomina *tubo*.

El entrenamiento se realiza modelando los movimientos de la muchedumbre en la escena mediante una colección de HMMs (Modelos ocultos de Markov). Los estados ocultos codifican los múltiples movimientos posibles que pueden ocurrir en cada posición espacial. Las probabilidades de transición de los HMMs definen las distintas dinámicas de variación temporal de los objetos en movimiento (**Fig. 5**). Así es posible codificar espacial y temporalmente las variaciones de movimiento de los objetos (individuos) en la escena.

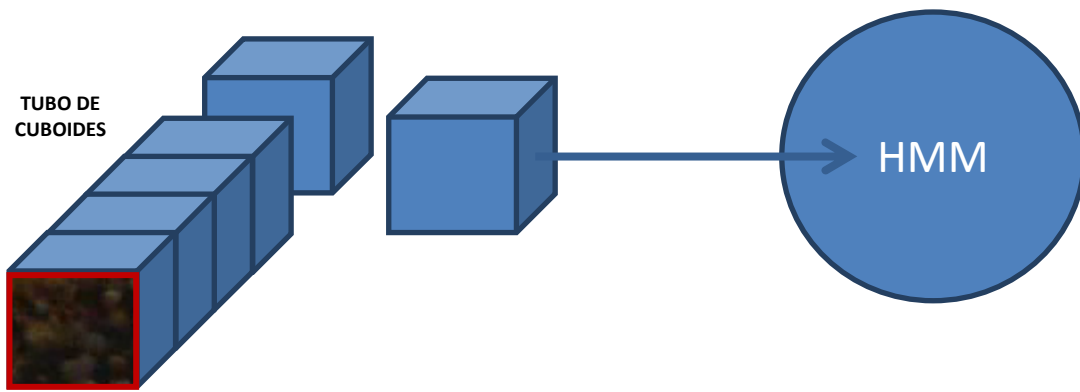


Fig. 4. Entrenamiento de cada tubo del vídeo, que representa la secuencia temporal de cuboides en la misma zona de la imagen. El entrenamiento se realiza con cada uno de estos cuboides, obteniendo un modelo para cada una de las zonas de la imagen.

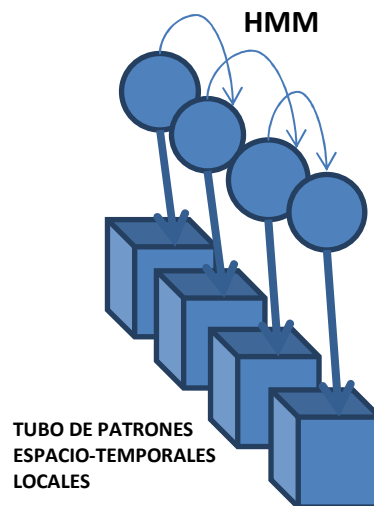


Fig. 5. Resultado del entrenamiento de una de las zonas de la imagen. El HMM resultante hace referencia en cada estado a una serie de patrones que definen los diferentes movimientos de los objetos en la zona.

Una vez se ha realizado el entrenamiento utilizando secuencias de vídeo representativas de los flujos de movimiento en la escena de interés y se han obtenido los HMMs de los diferentes tubos, se utilizan estos patrones para que, dada una serie de observaciones en una nueva secuencia de vídeo, se pueda predecir el patrón de movimiento en cada zona de las imágenes (3). Este patrón predicho es utilizado posteriormente para mejorar la efectividad de un algoritmo de seguimiento basado en filtro de partículas que permite realizar el seguimiento de objetos específicos de la escena.

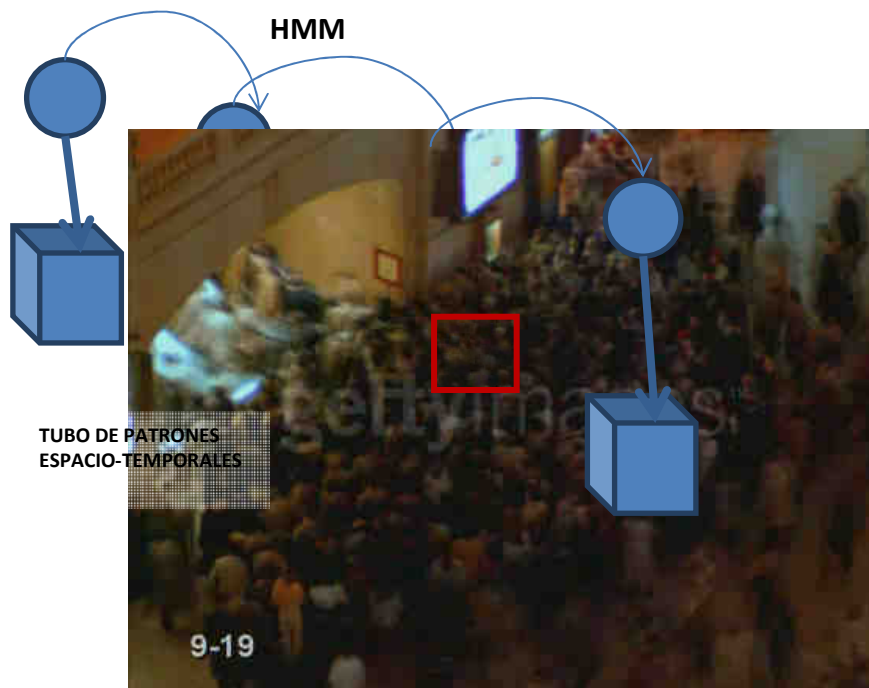


Fig. 6. Predicción del patrón espacio-temporal posterior dada una serie de observaciones anteriores y el HMM entrenado previamente.

En concreto, la predicción de patrones de movimiento se usa posteriormente para determinar la distribución probabilística de direcciones predominantes de flujo óptico en la región o regiones que contienen el objeto que se está siguiendo. Con estas distribuciones de flujo óptico, el algoritmo de seguimiento basado en filtro de partículas es capaz de actualizar la población de partículas teniendo en cuenta las direcciones más probables de movimiento en la escena analizada, lo cual redundará en un seguimiento con mayor efectividad que cuando no se dispone de dicha información previa.

Modelo espacio-temporal

El modelo espacio-temporal que se ha desarrollado está basado en regiones locales. Se utilizan estas regiones para asimilar la variabilidad en el movimiento de la muchedumbre dentro de un proceso de aprendizaje. Para esto se ha seguido el trabajo previo (4) de los mismos autores, desechando otros modelos alternativos que no tienen en cuenta relaciones temporales entre movimientos locales secuenciales (2).

Para la obtención de los modelos o patrones iniciales es necesario representar de forma estadística el movimiento. Para ello se particiona la

secuencia de vídeo en cuboides que constan tanto de una componente espacial como de una componente temporal. Estos cuboides no se deben solapar (**Fig. 3**). Los cuboides en la aplicación que se ha desarrollado son configurables y dependen de tres valores: alto, ancho y número de cuadros de la secuencia de vídeo. El alto y el ancho son los que confieren la dimensión espacial, mientras que el número de cuadros proporciona la dimensión temporal.

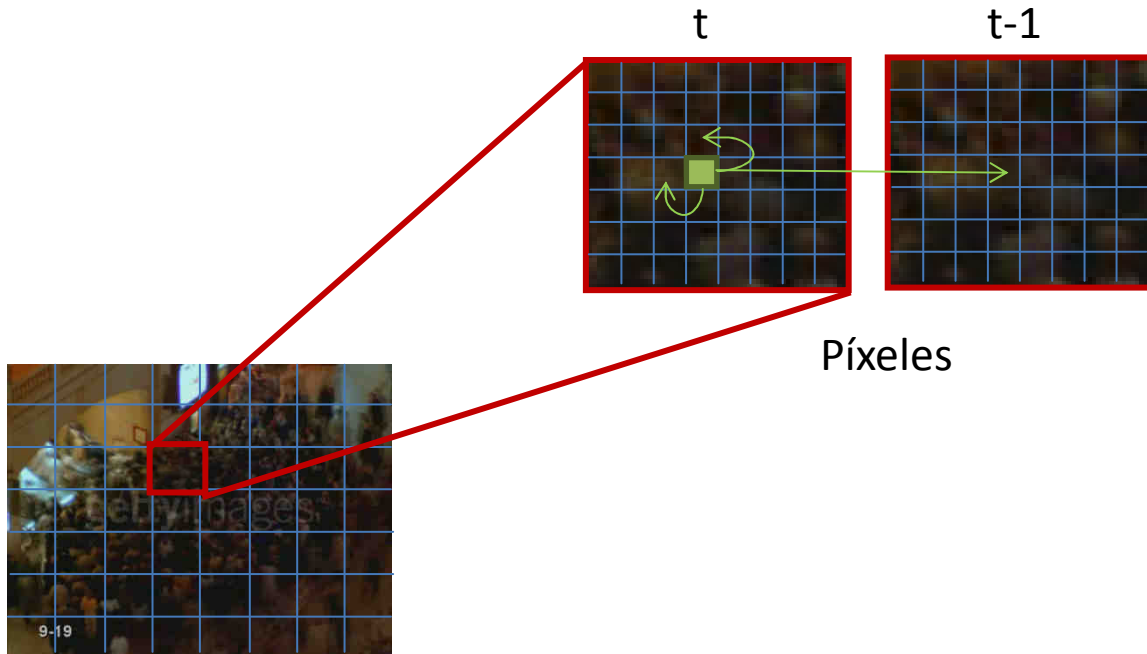
Cada cuboide tiene asociada una distribución Gaussiana 3D $\mathcal{N}(\mu, \Sigma)$ definida por una media y una matriz de covarianzas:

$$\mu = \frac{1}{N} \sum_i^N \nabla I_i$$

$$\Sigma = \frac{1}{N} \sum_i^N (\nabla I_i - \mu)(\nabla I_i - \mu)^T$$

donde:

- N es el número de píxeles del cuboide.
- ∇I_i es el gradiente espacio-temporal calculado para cada píxel, que está constituido por las derivadas estimadas en la dirección horizontal (x), vertical (y) y temporal (t): $\nabla I_i = (I_x, I_y, I_t)$



Cuboides

Fig. 7. Cálculo del gradiente tridimensional. Se puede observar que se realiza la comparación con el píxel anterior horizontal, el píxel anterior vertical y el píxel anterior temporal.

Se obtienen así todos los patrones espacio-temporales para cada localización espacio-temporal del vídeo de entrenamiento. Con lo cual, en una localización espacial n y temporal t , el patrón espacio-temporal es O_t^n , que queda representado por μ_t^n y Σ_t^n .

En otros trabajos se han utilizado también distribuciones de gradientes (5), (6) y (7) o HoG (Histograms of oriented Gradients). Sin embargo, los HoGs se calculan sobre los gradientes espaciales, no utilizando la información temporal. En el algoritmo desarrollado se utiliza una distribución Gaussiana para representar la distribución estadística de los vectores del gradiente espacio-temporal.

Los patrones en una determinada posición de la imagen pueden cambiar de forma frecuente debido a que los objetos pueden moverse en muy diversas direcciones o, por ejemplo, pueden cambiar dependiendo del momento del día. Por esa razón se modelan esas dinámicas temporales de los gradientes espacio-temporales entrenando un Modelo Oculto de Markov (HMM) en cada región espacial del vídeo.

Los patrones de movimiento locales tienen una relación muy fuerte entre sí. Tanto si un objeto continúa moviéndose en la misma dirección como si cambia de dirección, se tiene en cuenta su movimiento para calcular los modelos de patrones generales de las regiones que ocupa en la escena. Estos cambios de dirección se reflejan en el HMM como transiciones entre estados. Las variaciones graduales en cambio no quedan reflejadas en el HMM.

La elección del modelo espacio-temporal final se realiza mediante un algoritmo de agrupamiento (*clustering*) en línea (4). Es decir, una vez se ha realizado el cálculo de todos los parámetros, μ y Σ , de cada cuboide, se recorren todos los cuboides de entrenamiento de cada tubo y se realiza una partición de dichos cuboides en clases diferentes. Cada nuevo cuboide de entrenamiento se compara con los anteriores ya clasificados. Si la diferencia entre los parámetros del cuboide y uno de los anteriores es más pequeña que un umbral configurable (al que llamaremos umbral Kullback-Leibler), el nuevo cuboide actualizará los parámetros correspondientes al cuboide previo. Si la diferencia fuera mayor, se introduce una nueva clase de cuboide, con su valor actual de los parámetros μ y Σ , pero sin realizar ninguna operación de actualización sobre los parámetros del resto de cuboides ya clasificados.

La función de actualización es la siguiente:

$$C_{final} = \frac{1}{N+1} C_{actual} + \frac{N}{N+1} C_{anterior}$$

donde:

- $C_{anterior}$ es el cuboide ya clasificado previamente cuya diferencia con el cuboide actual es menor que el umbral.
- C_{actual} es el cuboide actual que se está comparando con el resto.
- C_{final} es el cuboide resultante.
- N es el número de actualizaciones que ha tenido ya el patrón.

Esta operación de actualización se aplica sobre los dos parámetros que definen el patrón: la media de gradientes espacio temporales de todos los píxeles del cuboide, μ , y su matriz de covarianzas, Σ .

Para la comparación entre los dos cuboides se utiliza la divergencia Kullback-Leibler (8), que en caso de comparación de dos distribuciones

Gaussianas, f y g , con sus respectivas medias y matrices de covarianzas tiene la formulación siguiente:

$$KL(f, g) = \frac{1}{2} \text{trace} \left\{ (\Sigma_f^{-1} + \Sigma_g^{-1})(\mu_f - \mu_g)(\mu_f - \mu_g)^T + (\Sigma_f + \Sigma_g^{-1}) + (\Sigma_g + \Sigma_f^{-1}) + 2I \right\}$$

donde:

- $KL(f, g)$ es la función de divergencia Kullback-Leibler, que devuelve 0 si ambas distribuciones son iguales.
- trace es la función traza de una matriz.
- I es la matriz identidad.
- f y g son dos distribuciones Gaussianas con medias y covarianzas μ_f, Σ_f y μ_g, Σ_g respectivamente.

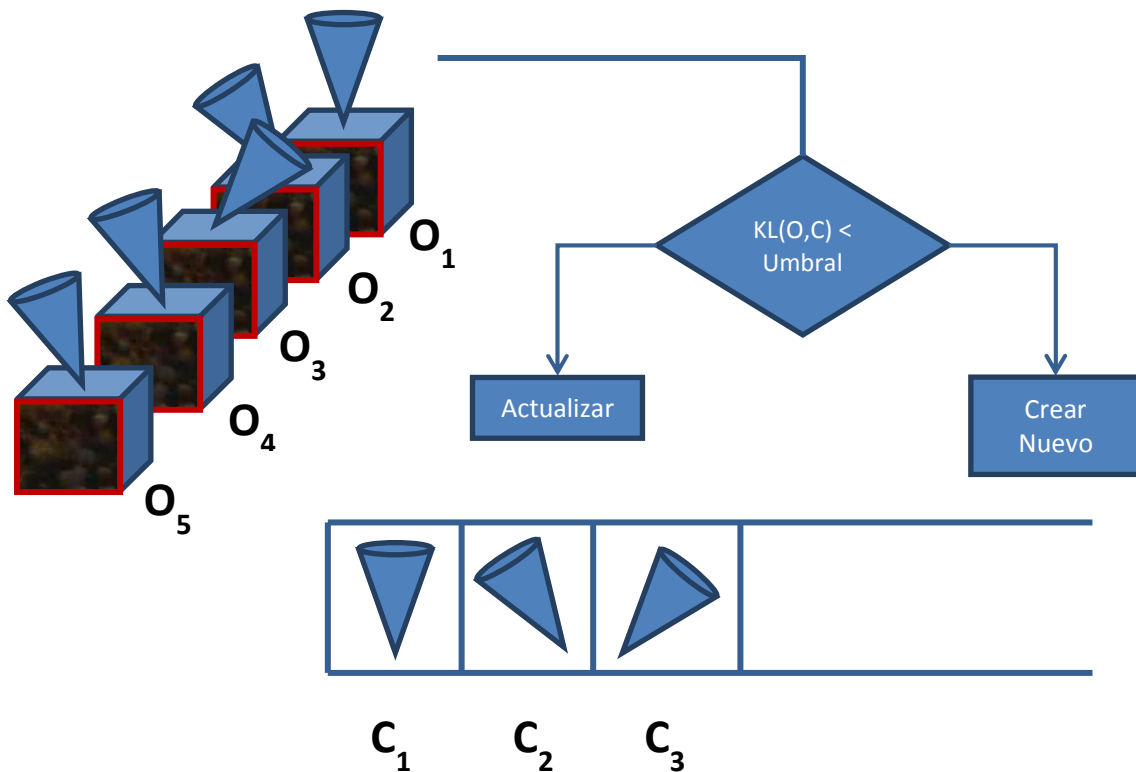


Fig. 8. Proceso inicial de *clustering online*. Cada una de las observaciones O en forma de cuboide se va insertando en una lista. Si la diferencia entre los cuboides es mínima, se actualiza el patrón clasificado y no se introduce una nueva.

Después de este proceso se obtiene una lista con la clasificación de los distintos estados del HMM, pero estos no son los valores que se utilizarán

como valores clasificados. Hasta ahora el proceso realizado ha servido para obtener una lista de clasificaciones, pero los valores finales de μ y Σ se calculan de este modo para terminar obteniendo cada patrón esperado: $P_s = \{\mu_s, \Sigma_s\}$

$$\mu_s = \frac{1}{N_s} \sum_i^{N_s} \mu_i$$

$$\Sigma_s = \frac{1}{N_s} \sum_i^{N_s} (\Sigma_i + \mu_i \mu_i^T) - \mu_s \mu_s^T$$

donde:

- N_s es el número de patrones dentro de cada clúster clasificado en el paso previo.

Una vez finalizado este paso, obtenemos todos los estados del HMM para cada tubo de la imagen. Cada uno de estos estados representa un patrón de movimiento prototipo de esa zona de la secuencia.

El resto de parámetros necesarios para el HMM se obtienen utilizando el algoritmo Baum-Welch (9). La inicialización de las matrices que utiliza el algoritmo Baum-Welch se realiza de la siguiente forma: la matriz de transición otorga la misma probabilidad a todas las transiciones, por lo que todos los valores de la matriz son iguales. En el caso de la matriz de inicialización, se considera que el algoritmo comienza en el primer estado, por lo que la probabilidad de inicialización de ese estado será 1 y el resto 0. Por último, la matriz de emisión es inicializada con la matriz identidad.

Una vez ejecutado el algoritmo Baum-Welch, se obtienen las tres matrices mencionadas previamente con los datos del entrenamiento. Estas matrices codifican los múltiples movimientos que pueden existir en cada localización espacial y temporal.

Predicción

Una de las ventajas de estar basados en un paradigma centrado en la escena y no en el objeto, es que la predicción es independiente del individuo u objeto al que se desea seguir. De hecho, la predicción a realizar será global

para toda la escena. Esto se traduce en que se calcula la predicción para cada tubo de la escena en un determinado momento.

Dado el HMM de uno de los tubos y una secuencia de patrones observados, se predice el siguiente patrón espacio-temporal en el instante siguiente de tiempo t . La distribución predictiva es la siguiente:

$$p(O_t|O_1, \dots, O_{t-1}) = \sum_{s \in S} p(O_t|s) \sum_{s' \in S} p(s|s') \hat{\alpha}_{t-1}(s')$$

donde:

- $p(O_t|O_1, \dots, O_{t-1})$ es la probabilidad de que dada una serie de observaciones se obtenga la observación O_t .
- $p(O_t|s)$ es la probabilidad de que dado un estado se produzca la observación O_t , es decir, la matriz de emisión.
- $p(s|s')$ es la matriz de transición de estados y s y s' representan los distintos estados.
- $\hat{\alpha}_{t-1}(s')$ es el vector de mensajes escalados obtenido del algoritmo *forward-backward*.

Por lo tanto, es necesario utilizar el algoritmo *forward-backward* para obtener el parámetro $\hat{\alpha}_{t-1}(s')$ (9). Para realizar este cálculo se necesitan las matrices calculadas previamente en el entrenamiento.

Se define el sumatorio de la ecuación anterior como $\gamma(s)$, que combina la distribución de transición entre los estados específicos y las probabilidades posteriores $\hat{\alpha}(s')$:

$$\gamma(s) = \sum_{s' \in S} p(s|s') \hat{\alpha}(s')$$

donde:

- s y s' son estados posterior y anterior respectivamente.

Se define \tilde{O}_t como el patrón predicho en el instante de tiempo siguiente t . Si se utiliza la fórmula anterior, el valor esperado viene definido por la distribución predictiva anterior:

$$\tilde{O}_t = \mathbb{E}[p(O_t|O_1, \dots, O_{t-1})] = \sum_{s \in S} \mathbb{E}[p(O_t|s)] \gamma(s)$$

El valor esperado de probabilidad de emisión $p(O_t|s)$ es el mismo patrón de movimiento $P_s = \{\mu_s, \Sigma_s\}$, por lo que el patrón espacio-temporal predicho es una

suma ponderada del conjunto de distribuciones Gaussianas asociadas con los estados ocultos del Modelo Oculto de Markov:

$$\tilde{\mu}_t = \sum_{s \in S} \gamma(s) \mu_s$$

$$\tilde{\Sigma}_t = -\mu_t \mu_t^T + \sum_{s \in S} \gamma(s) (\Sigma_s + \mu_s \mu_s^T)$$

donde:

- μ_s y Σ_s son la media y matriz de covarianzas del estado oculto respectivamente.
- $\tilde{O}_t = \{\tilde{\mu}_t, \tilde{\Sigma}_t\}$ es el patrón predicho.

Una de las ventajas de estas predicciones es que su cálculo es muy eficiente, por lo que, una vez realizado el entrenamiento, la obtención de los patrones de predicción se puede realizar en tiempo de ejecución.

Seguimiento

Una vez obtenido el patrón espacio-temporal de movimiento predicho, éste puede incluirse como parámetro dentro de un algoritmo de seguimiento basado en filtro de partículas. El modelo espacio-temporal creado en los HMMs en el transcurso de la secuencia puede mejorarse y variar en el tiempo para así ir adaptándose a los nuevos movimientos que se puedan producir por parte de los individuos u objetos en la escena.

Con los datos obtenidos previamente, se impone una distribución de transición utilizando los patrones espacio-temporales predichos. Por eficiencia computacional, se utilizan los datos del cuboide que ocupa el lugar del centro de la región de la que se desea hacer el seguimiento (**Fig. 9**).

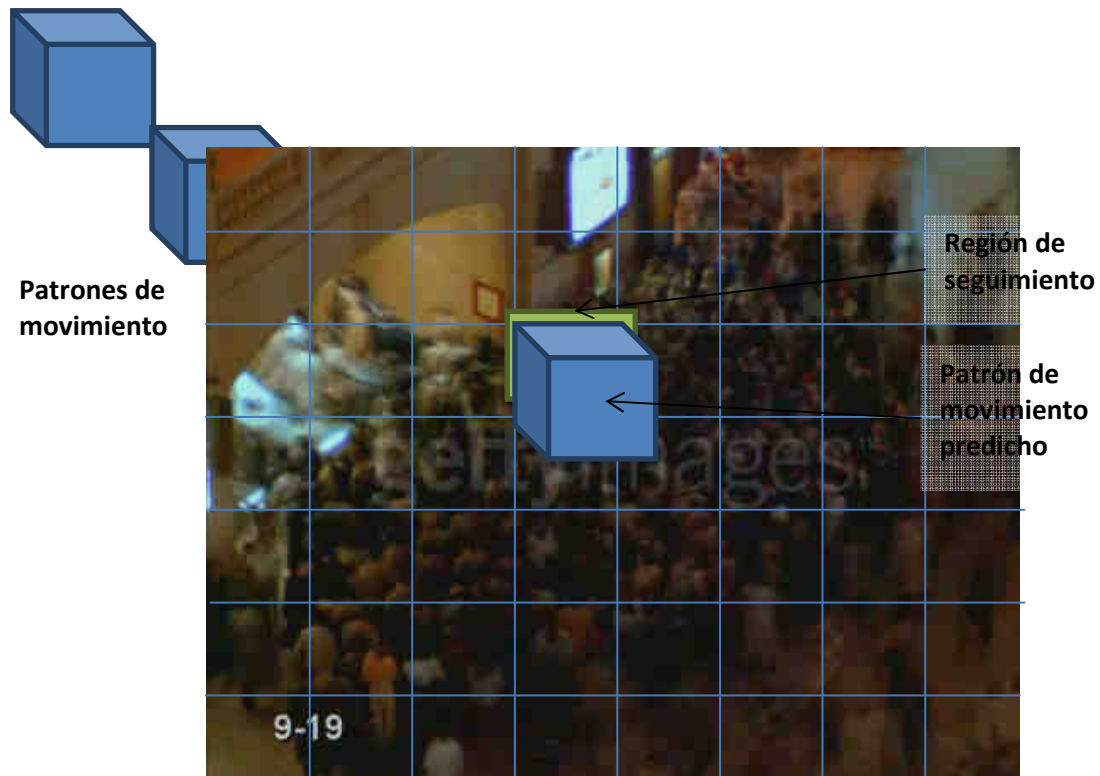


Fig. 9. Proceso de generación del patrón predicho para cada cuboide y selección de la región de acción. La región que se utiliza para los cálculos es aquella en la que el centro de la región de seguimiento se encuentre. En la figura se observa que para este caso concreto, la región seleccionada es sobre la que se encuentra el cuboide y la región verde la región a seguir, cuyo centro se encuentra en la región del cuboide.

Se usa el patrón espacio-temporal predicho para realizar una hipótesis del movimiento del objeto a seguir entre los cuadros $f - 1$ y f , esto es, el cuadro anterior y el cuadro actual, obteniendo la distribución de transición $p(x_f|x_{f-1})$. Siendo:

$$x_f = [k_f^T, d_f^T]$$

donde:

- x_f es el vector de estado.
- $k_f = [x, y]$ es la localización del objeto a seguir.
- $d_f = [w, h]$ es la caja que contiene el objeto a seguir, con un ancho y un alto (w y h) respectivamente.

La distribución de transición de la localización del objetivo a seguir $p(k_f|k_{f-1})$ refleja un movimiento 2D del objetivo entre los cuadros $f - 1$ y f y se modela como una distribución Gaussiana definida de la siguiente forma:

$$p(k_f|k_{f-1}) = \mathcal{N}(k_f - k_{f-1}; \omega, \Lambda)$$

donde:

- ω es el vector del flujo óptico.
- Λ es la matriz de covarianzas.

Los parámetros ω y Λ se estiman a partir del patrón espacio-temporal predicho en la región de la escena que contiene k_{f-1} . El patrón que calculamos previamente $\tilde{O} = \{\tilde{\mu}, \tilde{\Sigma}\}$ se basa en una media y una matriz de covarianzas. En (10) y (11) se estudia la relación entre los gradientes espacio-temporales, específicamente en forma de matriz de estructura tensorial:

$$\tilde{G} = \tilde{\Sigma} + \tilde{\mu}\tilde{\mu}^T$$

De aquí podemos obtener el flujo óptico según se indica en (10):

$$\tilde{G}w = 0$$

donde:

- $w = [u, v, z]^T$ es el vector de flujo óptico 3D definido en coordenadas homogéneas.

Como el cuadro en el que se desea realizar el seguimiento es el siguiente en el tiempo, la dimensión z se elimina para obtener un vector de flujo óptico 2D que permite realizar la búsqueda del objeto a seguir en el espacio de imagen:

$$\omega = \left[\frac{u}{z}, \frac{v}{z} \right]^T$$

El vector de flujo óptico 3D es el vector propio de \tilde{G} que tiene el valor propio más pequeño (11). Para esto se resuelven los valores propios y vectores propios de \tilde{G} , ordenándolos de menor a mayor valor propio, y se obtiene el vector propio con valor propio menor.

Para estimar la matriz de covarianzas Λ , se utiliza también la relación entre los gradientes espacio-temporales. En concreto, el flujo óptico w es ortogonal al plano que contiene los gradientes espacio-temporales.

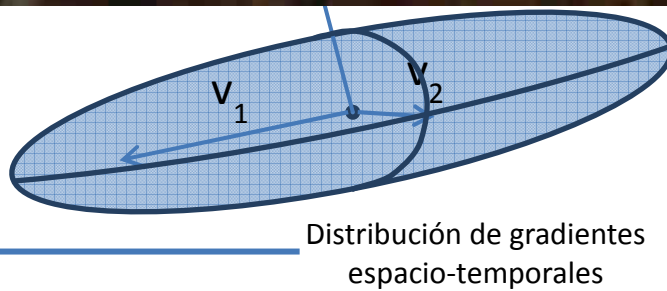


Fig. 10. Ortogonalidad del flujo óptico 3D en la imagen en el cuadro $f=1$. La distribución de gradientes espacio-temporales se proyecta en la imagen 2D obteniendo el flujo óptico 2D. Los otros dos vectores propios representan la variación del flujo óptico.

Los valores propios obtenidos de \tilde{G} son λ_1 , λ_2 y λ_3 . Estos valores estarán ordenados y $\lambda_3 \ll \lambda_1, \lambda_2$. Con el vector propio relativo al menor de ellos se obtiene el flujo óptico 2D. Es decir, si el vector propio obtenido es $[u, v, z]^T$, ese será el flujo óptico 3D y la proyección será el flujo óptico 2D.

Los otros dos vectores propios restantes v_1 y v_2 representan la variación del flujo óptico y de igual forma se pueden proyectar en el plano $f = 1$. Las proyecciones son v_1' y v_2' . Con estas proyecciones y los valores propios se realiza el cálculo de la matriz de covarianzas:

$$\Lambda = [v'_1, v'_2] \begin{bmatrix} \frac{\lambda_3}{\lambda_1} & 0 \\ 0 & \frac{\lambda_3}{\lambda_2} \end{bmatrix} [v'_1, v'_2]^{-1}$$

Estos dos parámetros, el flujo óptico y la matriz de covarianzas del flujo óptico, son los dos parámetros necesarios para realizar un seguimiento correcto del objeto. Estos dos parámetros son los que se incorporan al algoritmo de seguimiento basado en filtro de partículas (12) en la etapa de actualización de la población de partículas.

El problema del seguimiento en los entornos masificados es la gran similitud entre diversas regiones de la escena, por lo que la utilización del flujo óptico predicho para cada región y su matriz de covarianzas permiten reducir el número de posibles regiones donde el algoritmo debe realizar la búsqueda del objetivo.

El funcionamiento de un algoritmo de seguimiento mediante filtro de partículas típico comienza con la selección de una región de seguimiento que define el objeto o individuo a seguir durante la secuencia. De esta región se obtiene alguna característica para poder comparar con las distintas partículas. En este caso se usan histogramas de color para realizar la comparación.

El número de partículas normalmente viene definido previamente y suele ser configurable. En referencia a esto, hay que tener en cuenta que un número pequeño de partículas reduce la cantidad de posibilidades de seguimiento en la imagen y un número muy alto aumenta considerablemente el tiempo de ejecución del algoritmo, lo que unido a regiones de seguimiento muy grandes o vídeos con mucha resolución, puede resultar en un retraso del algoritmo y la pérdida de la condición de ejecución en tiempo real.

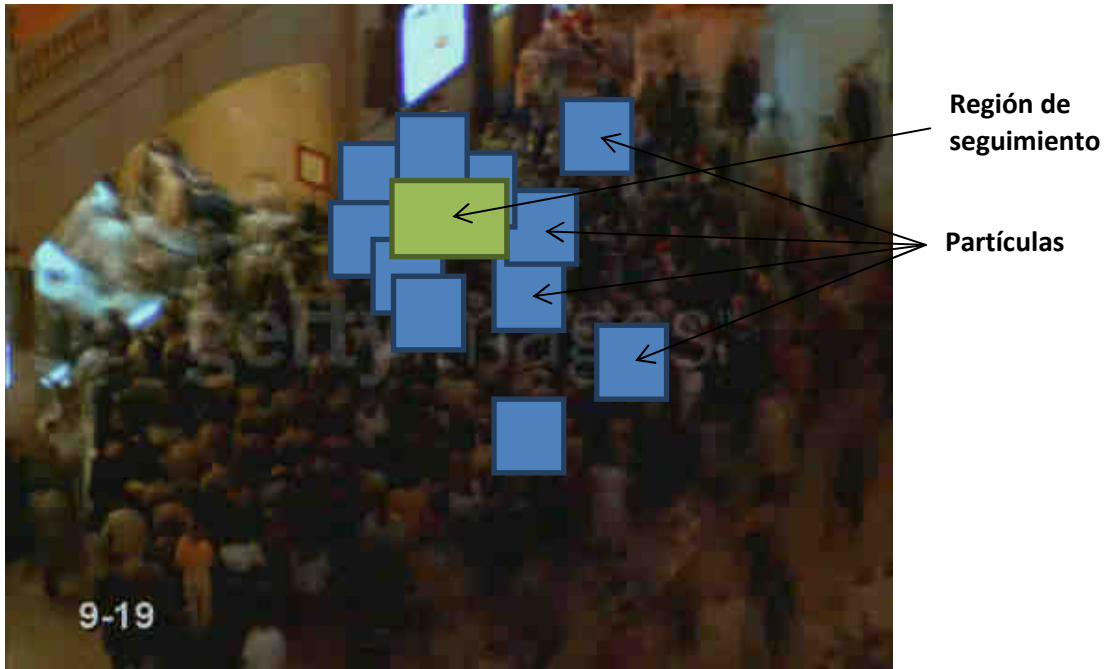


Fig. 11. Cálculo de partículas posibles (en azul) mediante un filtro de partículas sin el uso del flujo óptico predicho ni su matriz de covarianzas. En verde, la región de seguimiento es el objeto que se desea seguir. Se puede observar que las partículas se distribuyen con una distribución Gaussiana.

En el algoritmo original de seguimiento basado en filtro de partículas, el cálculo de la posición de las partículas posibles se realiza mediante una distribución Gaussiana alrededor del punto donde se encontraba la región de seguimiento en el instante anterior $t-1$. Mediante esta Gaussiana, se obtienen los puntos centrales de las partículas que después se compararán con la región de seguimiento que contiene el objeto a seguir:

$$p_i = [x_t, y_t]$$

$$x_t = x_{t-1} + gaussian(SD)$$

$$y_t = y_{t-1} + gaussian(SD)$$

donde:

- i representa la partícula y p_i el punto central o de anclaje de esta partícula.
- x_t, y_t son las coordenadas del punto central de la partícula que se calcula.
- x_{t-1}, y_{t-1} son las coordenadas del centro de la región de seguimiento.

- $gaussian(SD)$ es una función que devuelve un valor que sigue una distribución Gaussiana característica y SD es una constante para inicializar esta Gaussiana.

En el caso que nos concierne, el seguimiento se realizará utilizando el flujo óptico y la matriz de covarianzas calculadas para estimar el centro de las nuevas partículas.

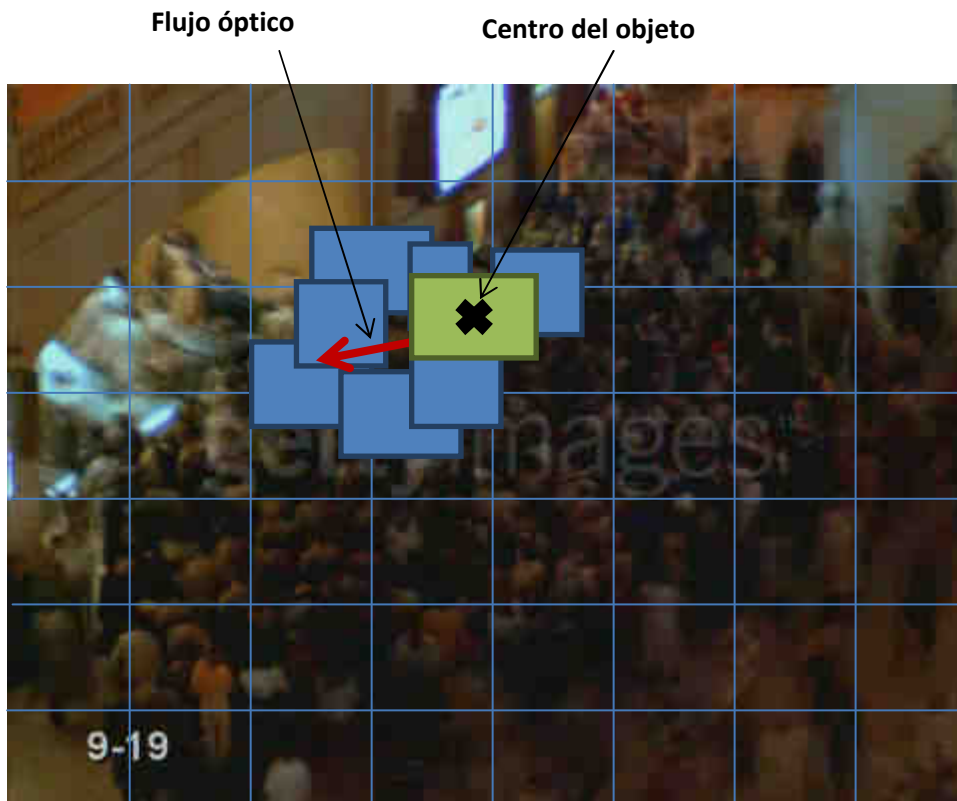


Fig. 12. Proceso de cálculo de partículas del filtro de partículas. Se busca el centro del objeto a seguir, se obtiene el flujo óptico predicho de la región en la que se encuentra el centro y se aplica el flujo óptico y la matriz de covarianzas para hallar las nuevas posiciones de las partículas que, en este caso, sí estarán ligadas al movimiento de los objetos aprendido en las secuencias de entrenamiento.

Para obtener las nuevas posiciones de las partículas, ahora al punto central se le agregará el flujo óptico y se le permitirá una variación dependiendo de la función Gaussiana utilizando la matriz de covarianzas. Si se utilizan las ecuaciones generadas anteriormente:

$$p_i = [x_t, y_t]$$

$$\omega = [\omega_x, \omega_y]$$

$$\Lambda = \begin{bmatrix} \Lambda_{0,0} & \Lambda_{0,1} \\ \Lambda_{1,0} & \Lambda_{1,1} \end{bmatrix}$$

Se pueden describir las nuevas funciones para el cálculo de las nuevas partículas de la siguiente forma:

$$x_t = x_{t-1} + \omega_x + \text{gaussian}(\sqrt{\Lambda_{0,0}})$$

$$y_t = y_{t-1} + \omega_y + \text{gaussian}(\sqrt{\Lambda_{1,1}})$$

Por lo tanto, las nuevas partículas aparecerán en la dirección del flujo óptico predicho con una dispersión que dependerá de la variabilidad de la matriz de covarianzas.

Con todas las nuevas partículas calculadas, se efectúa una comparación entre dichas partículas y la región de seguimiento donde se encuentra el objeto. Esta comparación se realiza directamente entre histogramas. Se utiliza para ello la distancia de Bhattacharyya (13), que permite comparar dos distribuciones de probabilidad, en este caso histogramas. En cada comparación se asigna un peso a cada partícula de manera que el peso es proporcional a la similitud entre histogramas. La mejor partícula es la de mayor peso, y es la candidata elegida en el proceso de seguimiento del objeto. En la siguiente iteración, esta partícula define la región de seguimiento y es la que se comparará con el resto de partículas.

Las partículas que quedan son actualizadas. La mitad de ellas, las de menor peso, quedan descartadas, mientras que la otra mitad se duplica. Esta copia es introducida dentro de la nueva colección de partículas, manteniendo el número total de partículas.

3. Detalles de la implementación

La implementación del algoritmo se ha realizado en C++ debido a su versatilidad y dado que se utilizan algunas librerías y aplicaciones desarrolladas en ese mismo lenguaje.

La librería utilizada para el procesamiento de la secuencia es OpenCV, en su versión 2.4. Esta librería es versátil en cuanto a contenidos y herramientas, y contiene un amplio conjunto de operaciones básicas con matrices e imágenes que facilitan el trabajo. Además, implementa algunas funciones matemáticas elaboradas, tales como el cálculo de valores y vectores

propios, lo que permite reducir significativamente el tiempo de ejecución de la aplicación.

La aplicación se ha separado en dos partes bien diferenciadas: el entrenamiento, y la predicción y seguimiento. Esto es debido a que el entrenamiento solo es necesario realizarlo una vez. Tras efectuar un entrenamiento completo y correcto, no es necesario repetir este entrenamiento de nuevo a no ser que cambien las condiciones de la escena (que aparezcan nuevos obstáculos fijos, que cambien los patrones de movimiento característicos de los objetos por cambios de rutinas o de horarios, etc...).

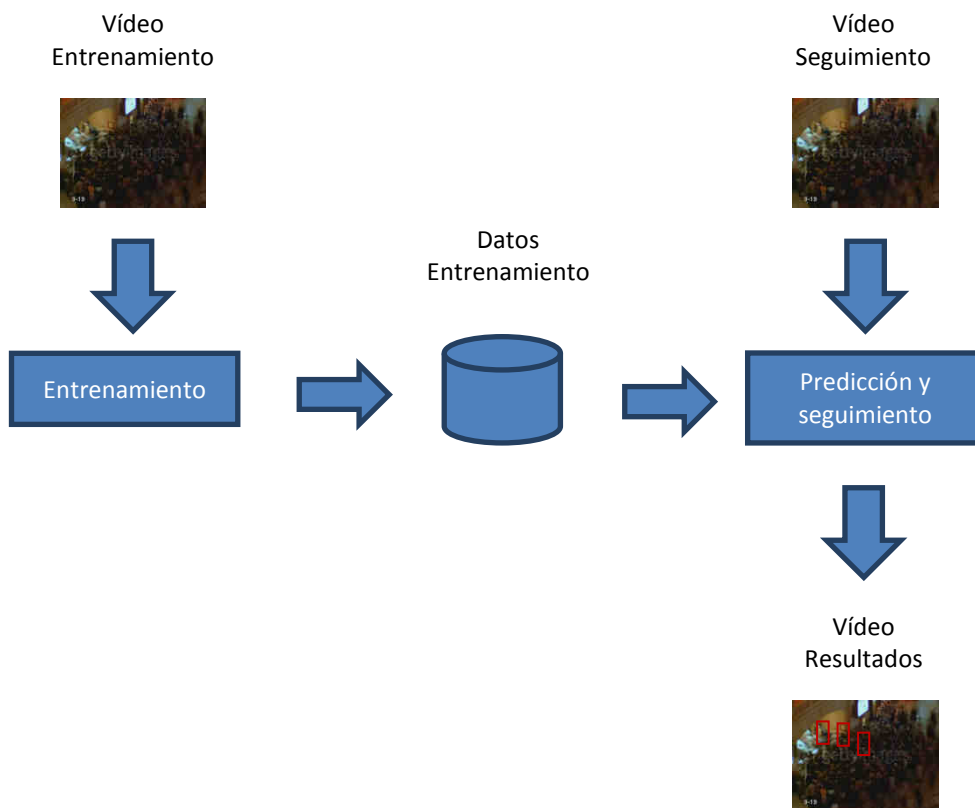


Fig. 13. Organización de la aplicación. Por una parte, la aplicación de entrenamiento genera unos datos a partir de la secuencia de vídeo de entrenamiento. Estos datos son los que más adelante se introducen en la aplicación de predicción y seguimiento junto con la secuencia de vídeo en la que se desea hacer el seguimiento, generando los resultados finales.

Los datos generados en la fase de entrenamiento son: el número de tubos o regiones en las que se divide la secuencia de vídeo, la lista de estados ocultos de Markov para cada una de estas regiones (con clasificación, matriz de medias y matriz de covarianzas de cada estado) y la lista de cuboides empleados en el entrenamiento con su clasificación. Por último, en los datos de

entrenamiento también se almacenan las matrices de transición, emisión e inicialización que definen los HMMs para cada región de la escena.

Para la realización de los HMMs se ha utilizado (14), una implementación de este algoritmo adaptada para la librería OpenCV. De igual forma, para el desarrollo del algoritmo de seguimiento basado en filtro de partículas se ha utilizado (15), una implementación del algoritmo para OpenCV sobre la que se han realizado las modificaciones necesarias para poder utilizar el código en el contexto de este trabajo.

Respecto a los tiempos de ejecución, el cálculo más costoso en la aplicación es el de la media y la matriz de covarianzas para todos los cuboides. Este cálculo consiste en la realización de una serie de operaciones matriciales repetidas para todos los píxeles de cada cuboide. En principio esto no sería un problema dado que el cálculo más complejo que hay que realizar de medias y matrices de covarianzas se efectúa en la fase de entrenamiento. En fase de ejecución este cálculo se realiza también en todos los cuboides, pero normalmente un cuboide puede ocupar en tiempo alrededor de 1 segundo de grabación (si son 25 cuadros por segundo, por ejemplo, y el número de cuadros por cuboide es de 25) y el cálculo de las matrices para 25 cuadros es viable realizarlo en 1 segundo.

4. Resultados experimentales

Para analizar el funcionamiento del algoritmo se han utilizado diversas secuencias de vídeo con el objetivo de comprobar qué capacidad de mejora se consigue utilizando el nuevo algoritmo.

Los vídeos utilizados se han obtenido del repositorio público “UCF Crowd Dataset” (16), que contiene una serie de secuencias de vídeo capturadas en entornos concurridos con diferentes casos de estudio característicos.

Los vídeos utilizados para este trabajo cumplen una serie de características: están capturados con cámara fija, representan entornos estables en los que no se producen cambios bruscos de movimientos ni situaciones anómalas excesivas, y no están afectados por cambios de luz en general.

Cabe destacar que la selección de los individuos u objetos a seguir se realiza de forma manual directamente sobre la imagen estática del vídeo. Una vez seleccionado un objetivo, el seguimiento se efectúa automáticamente.

En el primer vídeo se puede observar el caso de una multitud de personas entrando en una estación de metro. En este caso aparecen varias direcciones de movimiento predominantes en el flujo de individuos, que una vez aprendidos a partir de una secuencia de entrenamiento, permiten mejorar la capacidad de predicción del algoritmo de seguimiento.



Fig. 14. Vídeo que muestra la entrada a una estación de metro en hora punta. Se puede observar que las condiciones de luz no son buenas y la resolución tampoco, por lo que hacer un seguimiento teniendo en cuenta únicamente la intensidad puede llevar a situaciones inestables.

La resolución del vídeo es baja. Por tanto, los detalles del vídeo se difuminan. Por tanto, la apariencia del objetivo modelada mediante un histograma de color no es lo suficientemente distintiva como para garantizar un correcto seguimiento basado exclusivamente en la comparación de características visuales.

En concreto, la ejecución del filtro de partículas original para realizar el seguimiento de una zona sin detalles y con poca información diferenciadora del resto de la escena provoca un seguimiento inestable que salta de una posición a otra sin ningún control. Este tipo de comportamiento es habitual en secuencias como estas debido a dos factores principales: las constantes oclusiones, que producen un continuo cambio de las características visuales del objeto a seguir, y la similitud entre unas zonas y otras. Ambos factores confunden al algoritmo de seguimiento, que selecciona las partículas en relación a los pesos sin tener en cuenta los patrones de movimiento asociados con cada zona.



Fig. 15. Ejecución del algoritmo de filtro de partículas sin modificaciones. Se observa que el seguimiento es inestable y las partículas seleccionadas no corresponden con el camino correcto del individuo seguido. En la parte superior izquierda se indica el número del cuadro en el vídeo de seguimiento.

Para solucionar los problemas mencionados, el algoritmo de seguimiento tiene en cuenta los patrones de movimiento observados en la etapa de entrenamiento. En la **Fig. 16** se puede observar el comportamiento del filtro de partículas cuando se aplica la predicción de flujo óptico mediante la técnica implementada.



Fig. 16. Ejecución del algoritmo de seguimiento basado en filtro de partículas con los datos obtenidos en el entrenamiento. Se aprecia que el seguimiento, a pesar de las constantes oclusiones y la similitud entre las distintas zonas de la secuencia, mantiene la dirección correcta del objetivo.

El flujo óptico y su matriz de covarianzas predicha en cada momento guían al filtro de partículas a lo largo de la imagen, mejorando su capacidad de búsqueda del objeto seguido. Los parámetros utilizados para la ejecución del algoritmo en este ejemplo concreto han sido los siguientes: el alto y ancho de cada región de la cuadrícula es de 30 píxeles, y se consideran 25 cuadros por cuboide, de manera que cada cuboide corresponde a un intervalo temporal de un segundo. El filtro de partículas está configurado con 100 partículas.

El cambio del alto y ancho del cuboide afecta a nivel de organización de los movimientos. Si se eligen regiones muy pequeñas, se corre el riesgo de no detectar los diversos movimientos espacio-temporales debido a la poca

información con la que se cuenta en las mismas. Esto indicaría que los movimientos en esos puntos no pueden definir patrones de movimiento claros. Con un aumento del tamaño de las regiones de la cuadrícula se recogen más patrones de movimiento dentro del mismo dominio espacial. De cualquier forma, los movimientos quedan modelados por medio de los HMMs de las regiones de la secuencia, y el flujo óptico, como se observa en la **Fig. 17**, es similar. La ventaja de utilizar tamaños pequeños de región es que el vector de flujo óptico predicho para cada región representa mejor el movimiento de todos los píxeles contenidos en ella. En este ejemplo concreto, por cada vector de flujo óptico de una región de la imagen de la derecha se pueden generar 4 vectores de flujo óptico en la imagen de la izquierda.



Fig. 17. A la izquierda, flujo óptico en el cuadro 75 de la secuencia con cuboides de 25 píxeles de ancho y alto. A la derecha, flujo óptico en el cuadro 75 de la secuencia con cuboides de 50 píxeles de ancho y alto.

Cabe destacar que el tamaño apropiado del cuboide depende completamente del tamaño de las imágenes y de las propias características de la escena, por lo que es un parámetro susceptible de ser definido manualmente.

El número de partículas del filtro de partículas en entornos en los que no hay demasiada similitud entre distintas zonas de la escena ha de ser lo más alto posible sin sacrificar el tiempo de ejecución. De ese modo aumentan las posibilidades de encontrar el mejor resultado. Sin embargo, en el ejemplo que se está mostrando, en el que se dan patrones de movimiento relativamente homogéneos y un extensos, un elevado número de partículas puede crear una incertidumbre mayor dado que no solo se busca en la dirección del vector de flujo óptico predicho, sino también en otras zonas que se amplían cuanto mayor sea el número de partículas.



Fig. 18. Predicción y ejecución del algoritmo con 10 partículas en vez de 100. Se observa una mayor estabilidad en el resultado.

Por lo indicado antes, el número de partículas para este tipo de entornos en los que no hay una gran variabilidad de patrones de movimiento no es recomendable que sea muy elevado. Las pruebas experimentales llevadas a cabo con la aplicación que se ha desarrollado en este trabajo demuestran que 10 partículas son ya suficientes para realizar una búsqueda fiable de posibles candidatos.

En la siguiente secuencia de vídeo de ejemplo que se ilustra en la **Fig. 19**, los individuos van claramente en una dirección, por lo que el flujo óptico predicho también corresponde a esa dirección.

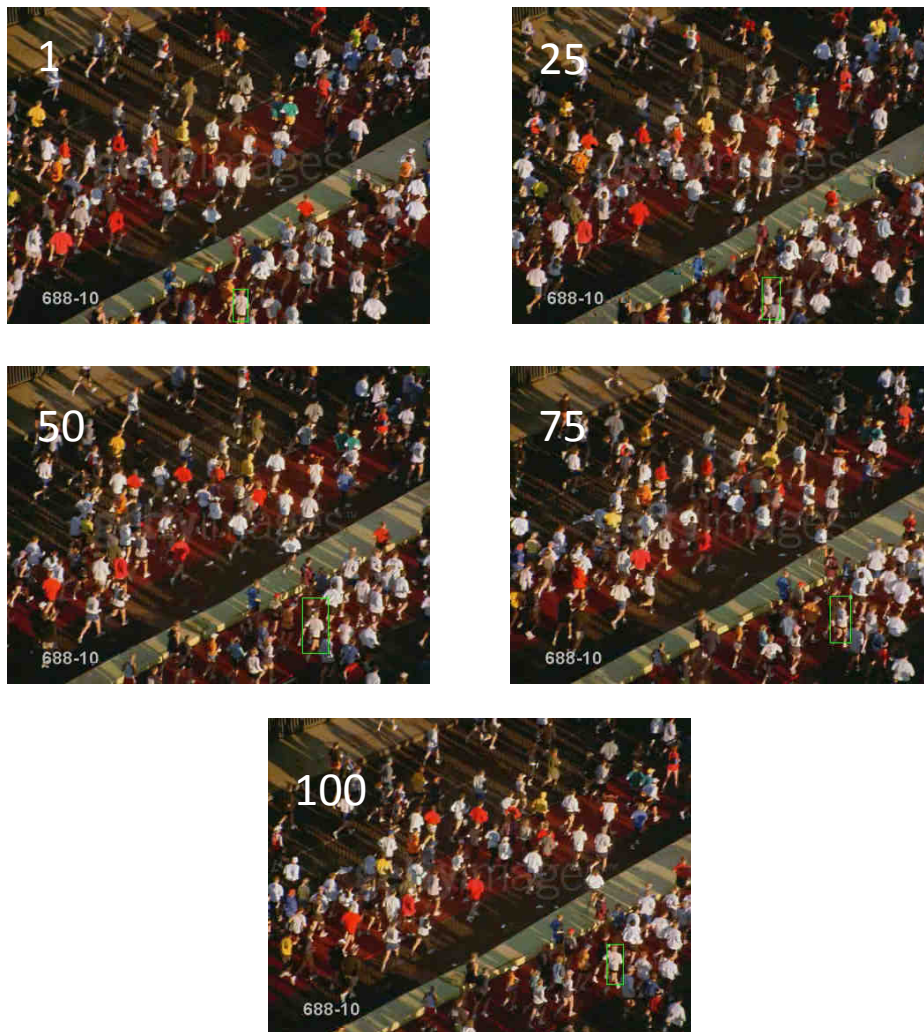


Fig. 19. Ejecución del algoritmo de seguimiento en la secuencia de vídeo de una carrera popular. Se puede apreciar el seguimiento de un corredor en la parte inferior de las imágenes.

El problema principal de este vídeo son las similitudes entre el corredor al que se está siguiendo y el resto de corredores. En el algoritmo de seguimiento basado en filtro de partículas este tipo de problemas se resuelve escogiendo al corredor más parecido al de la selección inicial. Sin embargo, debido a los movimientos que realiza el corredor, la apariencia varía y, por tanto, surgen otros corredores que en el instante siguiente tienen una apariencia más similar a la de dicha selección.

Otro de los parámetros importantes en la ejecución del algoritmo es el umbral asociado a la diferencia Kullback-Leibler (8) explicada anteriormente. Esta medida es la que determina las actualizaciones sobre los estados que se van generando en la etapa de agrupamiento (clustering) en línea, que es

crucial en la fase de entrenamiento del algoritmo desarrollado. De ese umbral depende el número de estados que se obtendrán para el entrenamiento de los HMMs correspondientes a cada región de la imagen.

Si se analizan los estados ocultos de Markov que se obtienen dependiendo del umbral de la diferencia Kullback-Leibler, se aprecia que la función resultante aumenta cuando el umbral de la diferencia Kullback-Leibler disminuye. Además, cuantos más estados ocultos se generan, mayor debe ser el tiempo de entrenamiento de los HMMs, aunque los resultados en ese caso son más precisos.

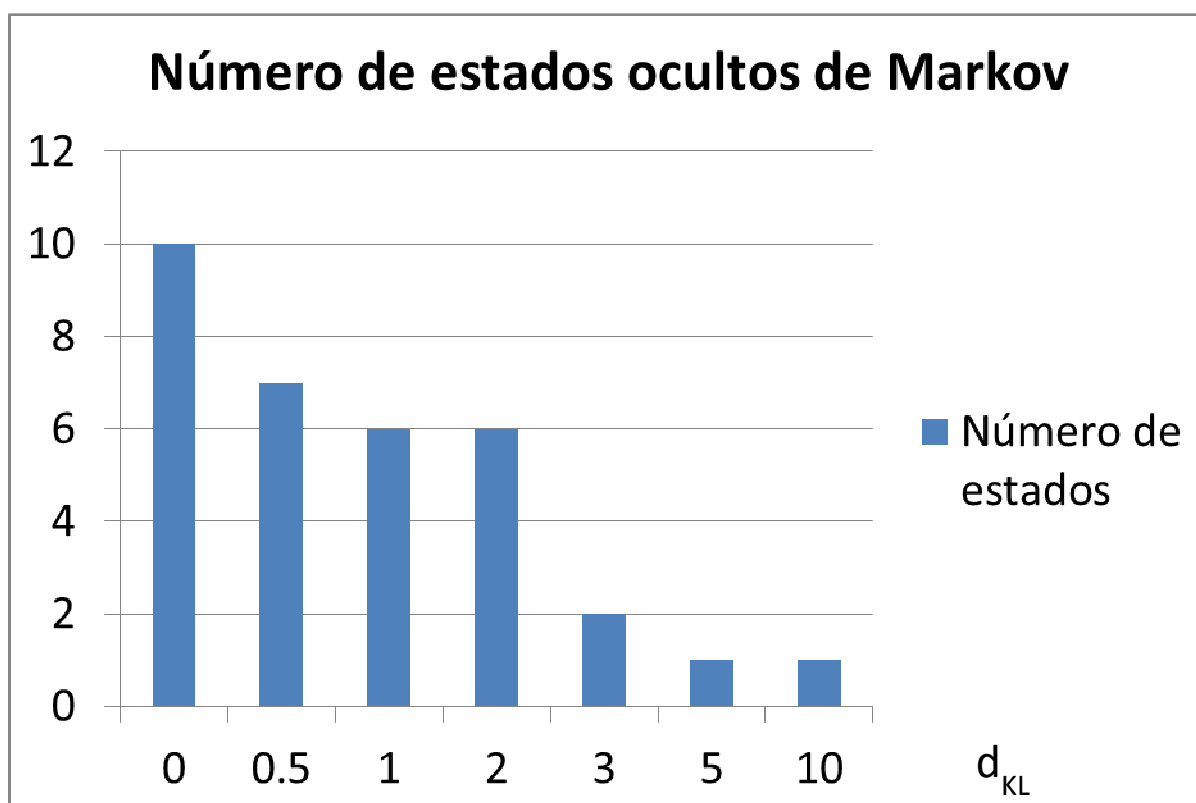


Fig. 20. Número de estados ocultos de Markov para distintos valores umbrales de la diferencia Kullback-Leibler. El número de estados es un factor importante que define la cantidad de patrones que se van a poder modelar mediante el HMM y qué peso tendrán las transiciones entre ellos. El equilibrio en el número de estados es vital para obtener unos resultados robustos y eficaces.

En el caso anterior, un umbral entre 1 y 2 es el más idóneo y el que proporciona una mayor estabilidad al algoritmo, ya que permite generar un número manejable de estados cuyas transiciones permiten modelar los distintos patrones de movimiento que suelen darse en las escenas habituales. Umbrales por encima de 3 solo dan lugar a un único estado por región, lo cual

es insuficiente para codificar la dinámica de movimientos dentro de dicha región.

5. Conclusiones

En este Trabajo de Fin de Máster se ha implementado uno de los algoritmos más relevantes del estado del arte en seguimiento de objetos en entornos concurridos, verificando su comportamiento y comparándolo con un algoritmo de seguimiento de propósito general. El algoritmo implementado tiene la capacidad de detectar y modelar automáticamente mediante modelos ocultos de Markov los patrones de movimiento más relevantes en una escena a partir de una secuencia de vídeo de entrenamiento. Utilizando los modelos aprendidos para cada región de la escena, el algoritmo es capaz de predecir la dirección de flujo óptico predominante en cada una de esas regiones dada cualquier secuencia de vídeo nueva de la misma escena, siempre que no hayan cambiado significativamente las direcciones de movimiento presentes en la secuencia o secuencias de entrenamiento.

Las direcciones de flujo óptico predichas permiten guiar la actualización de partículas de un algoritmo de seguimiento basado en filtro de partículas, lo cual conduce a unos resultados de seguimiento más precisos y estables que cuando no se dispone de información adicional y las partículas se actualizan aleatoriamente.

Los entornos muy concurridos de personas conllevan un elevado riesgo de propiciar situaciones peligrosas, entre las que se pueden destacar el pánico, estampidas y accidentes, que suelen involucrar a un gran número de individuos. La aplicación de técnicas como las desarrolladas en este trabajo puede tener un impacto significativo a la hora de minimizar los riesgos potenciales de este tipo de sucesos. Por ejemplo, el seguimiento de personas en grandes masificaciones puede facilitar la detección temprana de problemas como avalanchas.

Otra de las posibles aplicaciones de la técnica desarrollada es en los sistemas automáticos de detección de rutas, por ejemplo, en centros comerciales, donde sería posible la identificación en tiempo real de las rutas con mayor interés por parte de los usuarios, permitiendo así planificar, por ejemplo, estrategias comerciales o publicitarias.

Este tipo de algoritmos permiten predecir en tiempo real el flujo de la multitud. En entornos en los que el movimiento de la masa sigue patrones fijos de movimiento, estos algoritmos son altamente efectivos. Los modelos ocultos

de Markov, en concreto, no solo almacenan las direcciones principales de movimiento en cada región de la escena, sino que además modelan la dinámica completa del movimiento, incluyendo su velocidad.

Por otra parte, la predicción de patrones de movimiento a partir de modelos entrenados a priori se integra perfectamente en los métodos de seguimiento basados en filtros de partículas, aumentando así el potencial y la robustez de estas técnicas, que han demostrado un rendimiento notable en entornos reales.

A raíz del desarrollo de este trabajo y de su evaluación experimental se han observado una serie de ventajas e inconvenientes que se resumen a continuación.

Ventajas

Puesto que el aprendizaje y predicción de direcciones de movimiento se realiza de forma independiente para cada región de la imagen, se consigue una adaptabilidad a la escena que no consiguen otras técnicas que realizan un procesado único para toda el área de la imagen. Esto produce unos resultados de seguimiento más precisos y estables. Además, esto facilita la paralelización del proceso, pudiendo ejecutarse de manera simultánea en las distintas regiones, lo que permite la explotación efectiva de arquitecturas multiprocesador y multinúcleo, así como el procesado basado en GPUs.

El algoritmo tiene una fase de aprendizaje fuera de línea que genera así un conjunto de datos de entrenamiento. Esta es la parte más costosa del proceso. A partir de esta información generada automáticamente, la etapa de predicción en línea se ejecuta de manera muy eficiente, por lo que no tiene un impacto significativo en el tiempo de ejecución del algoritmo final de seguimiento.

Por último, al utilizar un paradigma centrado en la escena y no en el objeto, que permite predecir la dirección de movimiento de forma independiente en cada una de las regiones rectangulares en las que se particiona la escena, es posible aplicar la técnica implementada al seguimiento simultáneo de múltiples objetivos. En el caso de utilizar un algoritmo de seguimiento basado en filtro de partículas, basta replicar el proceso de seguimiento en tantas instancias como objetivos. El algoritmo de predicción se debe ejecutar una sola vez para cada nuevo cuboide (subconjunto de cuadros) de cada región de la escena. Los vectores de flujo óptico obtenidos y sus matrices de covarianzas respectivas pueden ser así utilizados para actualizar las partículas de cada uno

de los procesos de seguimiento activos, que pueden ejecutarse concurrentemente.

Inconvenientes

En algunas partes de la escena se pueden producir movimientos con direcciones sin un patrón claro. Esto provoca que el algoritmo desarrollado genere zonas muertas, sin dirección de flujo predominante, que se resuelven con el uso del filtro de partículas, aunque en este caso el algoritmo de predicción de movimiento no produce ningún tipo de mejora frente al filtro de partículas original.

Cada escena tiene unas propiedades diferentes y únicas, por lo que los parámetros aplicados al algoritmo han de variar en función de la secuencia de vídeo que se utilice. En algunos casos se puede utilizar un umbral de divergencia de Kullback-Leibler que en otros no es eficaz. De la misma forma, el resto de parámetros, como el ancho, alto y el número de cuadros de cada cuboide, dependerá de diversos factores, entre ellos del tamaño y resolución de las imágenes, del tamaño de los objetos que se desean seguir en la escena o de la cantidad de movimientos que se producen en la misma. Por tanto, el algoritmo de predicción ha de ser supervisado por un operador humano, que es el que deberá fijar dichos parámetros en función de los resultados que se pretendan alcanzar. Esta es, por otra parte, práctica habitual en los sistemas de monitorización y vídeo-seguridad actuales.

A pesar de que el problema de las oclusiones parciales queda en gran parte solucionado, las oclusiones completas seguirán impidiendo el seguimiento de los objetos en esas regiones. Si la oclusión es corta en el tiempo y no ocupa mucho espacio en la escena, el algoritmo de seguimiento podrá recuperarse, pero lo habitual es que ni siquiera los patrones de movimiento recojan este tipo de situaciones. En cualquier caso, el uso de un algoritmo que predice el movimiento en cada región de la escena es a priori beneficioso a la hora de volver a capturar un objetivo que estaba siendo seguido y que se perdió debido a alguna oclusión, siempre y cuando siga el movimiento predominante en esa región de la escena.

Mejoras

Teniendo en cuenta los fundamentos teóricos del algoritmo desarrollado y los resultados obtenidos, se proponen las siguientes mejoras, que abren la vía a futuras líneas de investigación que partan del trabajo desarrollado:

- Utilizar información de color. Consistiría en realizar el mismo proceso de entrenamiento y predicción de movimiento con los tres canales RGB o con canales HSI. Las predicciones obtenidas de manera independiente para cada canal se deberían fusionar convenientemente para generar un vector de flujo óptico predicho y una matriz de covarianzas asociada más robustos.
- Variar la etapa de comparación de partículas realizada por el algoritmo de seguimiento basado en filtro de partículas, de modo que en lugar de usar histogramas de color, se comparen directamente las medias y covarianzas de los cuboides asociados con dichas partículas. De esa manera la comparación tendría en cuenta no solo la apariencia visual, sino también la información espacio-temporal asociada a los cuboides.
- Utilizar información de movimiento de cuboides contiguos que, aunque con menos peso, pueden también colaborar en la generación del patrón de movimiento. Esto se puede realizar tanto en la fase de entrenamiento, donde el entrenamiento de una región podría influir en regiones contiguas, como en la fase de predicción y seguimiento, donde se podría utilizar no solo la información del cuboide más próximo al centro de la región de seguimiento, sino las regiones contiguas a este.

El problema del seguimiento de objetos en entornos concurridos plantea un enorme reto en el ámbito del análisis de secuencias de vídeo. Los grandes logros tecnológicos de los últimos años han puesto a disposición de todo el mundo equipos más rápidos y de mayor eficiencia energética, más recursos técnicos y, en definitiva, más información. Estas nuevas tecnologías permiten que algoritmos como el desarrollado en este Trabajo Fin de Máster puedan ser aplicables a nivel real y no ser solo meras propuestas teóricas sin recorrido práctico. Esto ha de redundar en la disponibilidad de nuevas capacidades que puedan acabar teniendo un impacto significativo en la sociedad.

6. Bibliografía

1. *Tracking Pedestrians using Local Spatio-temporal Motion Patterns in Extremely Crowded Scenes*. **Kratz, L. and Nishino, K.** IEEE Trans. on Pattern Analysis and Machine Intelligence. 2012, 34(5), pp. 987-1002.
2. *Tracking in Unstructured Crowded Scenes*. **M. Rodriguez, S. Ali, and T. Kanade.** 2009, Proc. of IEEE Int'l Conf on Computer Vision. 1389-1396.
3. *Tracking With Local Spatio-Temporal Motion Patterns in Extremely Crowded Scenes*. **Kratz, L. y Nishino, K.** 2010, Proc.of IEEE Int'l Conf on Computer Vision and Pattern Recognition. 693-700.
4. *Anomaly Detection in Extremely Crowded Scenes Using Spatio-Temporal Motion Pattern*. **Kratz, L. and Nishino, K.** 2009, Proc. of IEEE Int'l Conf on Computer Vision and Pattern Recognition, pp. 1446–1453.
5. *Histograms of Oriented Gradients for Human Detection*. **Dalal, N. y Triggs, B.** 2005, Proc. of IEEE Int'l Conf on Computer Vision and Pattern Recognition. 886 - 893.
6. *A Spatio-Temporal Descriptor Based on 3D-Gradients*. **Klaser, A., Marszałek, M. y Schmid, C.** 2008, Proc. of British Machine Vision Conf, 995–1004.
7. *Learning Realistic Human Actions from Movies*. **Laptev, I., y otros, y otros.** 2008, Proc. of IEEE Int'l Conf on Computer Vision and Pattern Recognition. 1-8.
8. *On Information and Sufficiency*. **Leibler, S. Kullback y A., R.** 1, 1951, The Annals of Mathematical Statistics, Vol. 22, pp. 79–86.
9. *A Tutorial On Hidden Markov Models and Selected Applications in Speech Recognition*. **Rabiner, L.** 77(2), Febrero 1989, Proc. of the IEEE, pp. 257–286.
10. *Space-Time Behavior Based Correlation*. **Shechtman, E. y Irani, M.** 2005, Proc. of IEEE Int'l Conf on Computer Vision and Pattern Recognition. 405–412.
11. *Analysis of Persistent Motion Patterns Using the 3D Structure Tensor*. **Wright, J. y Pless, R.** 2005, IEEE Workshop on Motion and Video Computing, 14–19.
12. *A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking*. **Arulampalam, S. M., Maskell, S. y Gordon, N.** 50, 2002, IEEE Transactions on Signal Processing, pp. 174-188.

13. *On A Measure of Divergence Between Two Statistical Populations Defined by their Probability Distributions*. **Bhattacharyya, A.** 1943, Bulletin of Cal. Math. Soc., Vol. 35, No. 1, pp. 99-109.
14. *Hidden Markov Model for OpenCV (CvHMM)*. **Sakhi, Omid B.** 2012, <http://sourceforge.net/projects/cvhmm/>.
15. *Particle filter tracking in OpenCV*. **Reyes, J.** 2011, <https://code.google.com/p/smallprograms/downloads/detail?name=pftracker.tar.gz>.
16. *A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis*. **Ali, Saad y Shah, Mubarak.** 2007, IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 1-6.