

Segmentación de objetos en movimiento utilizando GPCA, técnicas algebraicas y flujo óptico para aplicaciones en espacios inteligentes

Cristina Losada, Manuel Mazo, Sira Palazuelos

Departamento de Electrónica. Universidad de Alcalá
Alcalá de Henares, Madrid, España
losada@depeca.uah.es

Abstract — En este trabajo se aborda el tema de la segmentación de objetos en movimiento utilizando técnicas de Análisis de Componentes Principales Generalizado (G-PCA), junto con métodos basados en álgebra lineal para la obtención del flujo óptico. Para ello se propone, previa a la obtención del flujo óptico (y en consecuencia la segmentación), realizar una selección de los píxeles que son candidatos a pertenecer a objetos en movimiento. Posteriormente, sobre estos píxeles candidatos se aplican técnicas basadas en el álgebra lineal para la obtención del flujo óptico de cada uno de los objetos candidatos. Esta propuesta ha sido implementada en un “espacio inteligente” dotado de múltiples cámaras y se han realizado diversas pruebas. Dichas pruebas han puesto de manifiesto la validez de la propuesta y la notable mejora que supone, en cuanto a la precisión de la segmentación y el tiempo de procesamiento, si se compara con otras propuestas en las que se utiliza únicamente el algoritmo algebraico para la obtención del flujo óptico de los objetos dentro de la escena.

I. INTRODUCCIÓN

La segmentación de objetos en movimiento es una tarea fundamental para el análisis de secuencias de imágenes y consiste en agrupar la información presente en las mismas en conjuntos de píxeles cuyo movimiento en el plano imagen es coherente a lo largo de una secuencia. Y todo ello sin tener un conocimiento previo acerca de qué píxeles de la imagen se mueven de acuerdo a un determinado modelo de movimiento. El análisis de movimiento en secuencias de imágenes digitales tiene un gran interés para una amplia gama de aplicaciones de la visión artificial, tales como robótica móvil, monitorización de tráfico, vigilancia, registrado de imágenes, etc.

Existen en la literatura diferentes técnicas que tratan de analizar y extraer información sobre el movimiento de los objetos en una escena. Las primeras aproximaciones para segmentación de movimientos 2D son las basadas en discontinuidades en el flujo óptico [1], [2], pero estas técnicas presentan diferentes inconvenientes derivados tanto del problema de apertura, como de la presencia de ruido en las estimaciones del flujo óptico.

La segmentación del movimiento ha estado tradicionalmente ligada con la detección de movimiento, en la que cada región corresponde a un modelo particular de movimiento que explica los cambios temporales en dicha región de la imagen [3], [4]. También se han realizado diferentes propuestas para segmentación de movimiento basadas en *clustering* [5], [6]. Sin embargo, en cualquiera de los casos anteriores es necesario tener un conocimiento a priori del número de objetos en movimiento presentes en la escena, lo cual no siempre es posible.

En [7] y [8] se propone un algoritmo de segmentación de objetos con modelos de movimiento afín basado en la técnica matemática GPCA (*Generalized Principal Components Analysis*) que permite segmentar objetos en movimiento sin conocimiento previo del número de objetos presentes en la escena. Este algoritmo se describe brevemente en el siguiente apartado. Hay que indicar que esta técnica matemática está basada en el álgebra lineal pero en la literatura también se le conoce por PCA generalizado. En lo que sigue se diferenciaron, por tanto dos tipos de técnicas de Análisis de Componentes Principales Generalizado, uno que identificaremos por G-PCA [10] y otro por GPCA (este último también lo identificaremos como basado en el álgebra lineal). Hay que advertir que G-PCA propuesto en [10], a pesar de la coincidencia en el nombre, no guarda relación con la técnica GPCA de [7] y [8].

II. SEGMENTACIÓN DE MOVIMIENTOS AFINES 2D UTILIZANDO TÉCNICAS DE GPCA (ALGEBRA LINEAL)

Dado un conjunto de N medidas $\{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1}^N$ correspondientes a un número desconocido de movimientos afines, la técnica matemática GPCA permite obtener el número de movimientos n y los parámetros de cada movimiento afín $\{\mathbf{A}_i\}_{i=1}^n$, así como realizar la segmentación de las imágenes medidas en función de los modelos de movimiento estimados, siendo $\mathbf{y} = [I_{x_1} \ I_{x_2} \ I_t]^T \in \mathfrak{R}^3$ un vector que contiene las derivadas espaciales y la derivada temporal entre dos imágenes, y $\mathbf{x} \in \mathbb{P}^2$ las coordenadas homogéneas de los píxeles.

Cualquier medida en la imagen (\mathbf{x}, \mathbf{y}) asociada con algún movimiento debe satisfacer la restricción afín multicuerpo (1), que reduce el problema de segmentación a obtener el número de movimientos afines n y los parámetros de dichos movimientos a partir de la ecuación no lineal:

$$\varepsilon(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^n (\mathbf{y}^T \mathbf{A}_i \mathbf{x}) = 0 \quad (1)$$

Dada la estructura de esta ecuación, es posible utilizar la inmersión de Veronese de grado n , $v_n : \mathfrak{R}^3 \rightarrow \mathfrak{R}^{M_n}$ para reescribirla en forma bilineal:

$$v_n(\mathbf{y})^T \mathbf{A} v_n(\mathbf{x}) = 0 \quad (2)$$

siendo $\mathbf{A} \in \mathfrak{R}^{M_n \times M_n}$ la matriz afín multicuerpo que representa el producto tensorial simétrico de todas las matrices afines $\{\mathbf{A}_i\}_{i=1}^n$. La restricción anterior puede reescribirse:

$$(v_n(\mathbf{y}) \otimes v_n(\mathbf{x}))^T \mathbf{a} = \mathbf{L}_n \mathbf{a} = 0 \quad (3)$$

donde \otimes indica producto tensorial. La expresión (3) permite calcular el número de movimientos n imponiendo una restricción sobre el rango de la matriz \mathbf{L}_n de forma que el sistema de ecuaciones (3) tenga solución única. Conocido el valor de n , la matriz afín multicuerpo \mathbf{A} se obtiene resolviendo el sistema de ecuaciones lineales en \mathbf{a} .

Como se demuestra en [7], GPCA nos permite recuperar el flujo óptico $\{\mathbf{u}_i\}_{i=1}^n$ a partir del flujo óptico multicuerpo $\tilde{\mathbf{u}} = \mathbf{A} v_n(\mathbf{x})$ de forma sencilla con cualquiera de los algoritmos para hiperplanos propuestos en [9] y posteriormente recuperar los modelos de movimiento afín individuales, así como realizar la segmentación de movimiento.

Como ya se ha comentado, esta propuesta para segmentación de movimiento tiene algunas ventajas ya que no requiere un conocimiento previo del número de modelos de movimiento presentes en la escena (proporciona un método para calcularlo). Sin embargo, aunque el problema es no lineal, el algoritmo propuesto está basado en técnicas del álgebra lineal tras realizar el embebido de los datos mediante la inmersión de Veronese. Esto provoca que, a pesar de que la solución propuesta es algebraicamente correcta, la estimación de la matriz afín multicuerpo no es robusta en presencia de ruido en las medidas de la imagen.

En [7] se propone tratar las medidas ruidosas realizando un proceso de optimización que minimice una función del error algebraico definida por la restricción afín multicuerpo, mediante técnicas de minimización no lineal, utilizando el algoritmo algebraico para la inicialización.

Sin embargo, dada la naturaleza del problema, a medida que el ruido aumenta, el tiempo de convergencia del algoritmo también aumenta. Por otro lado, también es posible que la minimización devuelva una solución local, en lugar de la solución global del problema. Por esto no resulta adecuado para realizar la segmentación

de forma robusta, especialmente en casos en que existen restricciones de tiempo.

III. SELECCIÓN DE PÍXELES CANDIDATOS MEDIANTE ANÁLISIS DE COMPONENTES PRINCIPALES

A la vista de los problemas que presenta la solución anterior, derivados de la presencia de ruido en las derivadas espaciales y temporales (\mathbf{I}_{x1} , \mathbf{I}_{x2} e \mathbf{I}_t) obtenidas a partir de las imágenes captadas, y dado que en nuestro caso las imágenes de la escena son captadas por una cámara que permanece fija en el entorno en que se mueve el robot, se propone realizar una selección previa de aquellos píxeles que son candidatos a pertenecer a objetos en movimiento para, posteriormente aplicar el algoritmo definido en [7] y [8] pero únicamente sobre los píxeles seleccionados.

La selección de píxeles se realiza, mediante la obtención de un modelo de fondo, que luego se compara con cada una de las imágenes tomadas por la cámara utilizando técnicas de Análisis de Componentes Principales [9]. Dada la estructura del problema, se ha decidido utilizar el algoritmo de Análisis de Componentes Principales Generalizado [10] (G-PCA). La ventaja de G-PCA sobre el clásico PCA radica en que el primero emplea una representación matricial (en lugar de vectorial) de los datos, lo que permite conseguir una reducción en los requerimientos de memoria así como una disminución del tiempo de ejecución.

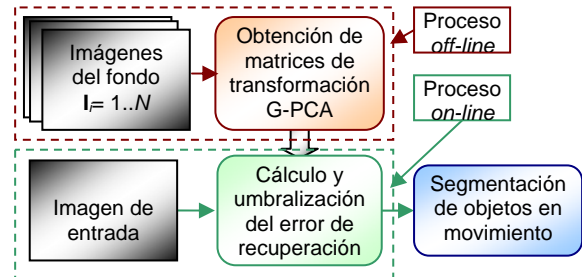


Fig. 1. Diagrama de bloques para la selección de píxeles candidatos

Como se puede observar en la Fig. 1, el algoritmo de selección de candidatos consta de dos etapas diferentes que se describen a continuación.

A. Obtención del modelo de fondo

En la primera etapa del algoritmo, la técnica G-PCA permite obtener dos matrices de transformación \mathbf{L} y \mathbf{R} a partir de un conjunto de imágenes del fondo $\{\mathbf{I}_j\}_{j=1}^N$, siguiendo los pasos representados en el diagrama de bloques de la Fig. 2, de forma que sea posible proyectar cualquier imagen del fondo al espacio transformado G-PCA, conservando las características principales de dicha imagen.

$$\mathbf{M} = \frac{1}{N} \sum_{j=1}^N \mathbf{I}_j \quad (4)$$

$$\tilde{\mathbf{I}}_j = \mathbf{I}_j - \mathbf{M} \quad j = 1..N \quad (5)$$

El primer paso para la obtención del modelo de fondo es estimar la media de las N imágenes (4) y a continuación restar a cada una de las imágenes la matriz calculada (5).



Fig. 2. Conjunto de imágenes del fondo de la escena.

Estas imágenes a las que se les ha sustraído la media son las que se utilizarán en el algoritmo, cuyas etapas se presentan en la Fig. 3, donde se puede observar que, tras la inicialización, se entra en un proceso iterativo que finaliza cuando se alcanza un criterio de convergencia del RMSE determinado por la variable η . La convergencia del algoritmo viene garantizada por el teorema 4.2 definido en [10].

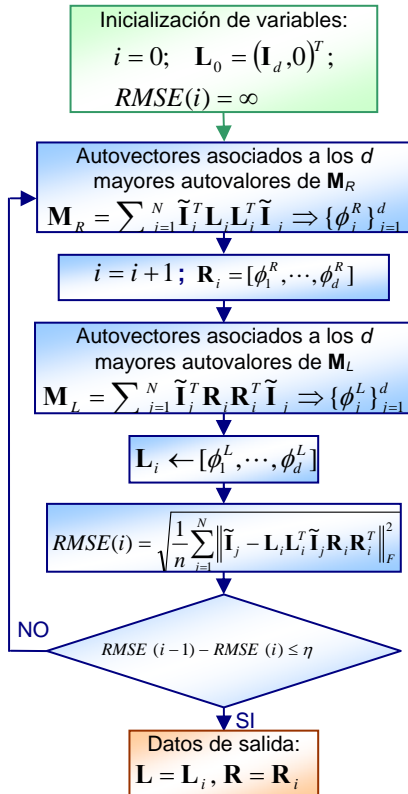


Fig. 3. Diagrama de bloques del proceso de obtención del fondo

También es importante el número de autovectores utilizados para obtener las matrices \mathbf{M}_R y \mathbf{M}_L , definidas según las ecuaciones (6) y (7):

$$\mathbf{M}_R = \sum_{j=1}^N \tilde{\mathbf{I}}_j^T \mathbf{L}_i \mathbf{L}_i^T \tilde{\mathbf{I}}_j \quad (6)$$

$$\mathbf{M}_L = \sum_{j=1}^N \tilde{\mathbf{I}}_j^T \mathbf{R}_i \mathbf{R}_i^T \tilde{\mathbf{I}}_j \quad (7)$$

donde \mathbf{L}_i y \mathbf{R}_i se actualizan en cada iteración del algoritmo, y dependen de los autovectores de las matrices \mathbf{M}_R y \mathbf{M}_L , ($\{\phi_j^R\}_{j=1}^d$ y $\{\phi_j^L\}_{j=1}^d$) obtenidas en la iteración anterior:

$$\mathbf{R}_i = [\phi_1^R, \dots, \phi_d^R] \quad (8)$$

$$\mathbf{L}_i = [\phi_1^L, \dots, \phi_d^L] \quad (9)$$

El número de autovectores d , coincide además con la dimensión del espacio transformado.

B. Selección de píxeles candidatos

En esta segunda etapa (Fig. 4) se compara el modelo de fondo obtenido con cada par de imágenes que se quiere segmentar para, de esta forma, determinar qué píxeles son candidatos a pertenecer a objetos que han entrado en la escena después de haber tomado las imágenes del fondo. Estos elementos pueden ser tanto robots controlados por el espacio donde se mueven (agentes controlados), como agentes con movimiento no controlado o, incluso nuevos elementos fijos, que se han introducido en la escena después de captar las imágenes utilizadas para obtener el modelo de fondo, y brillos o reflejos debidos a cambios en la iluminación.

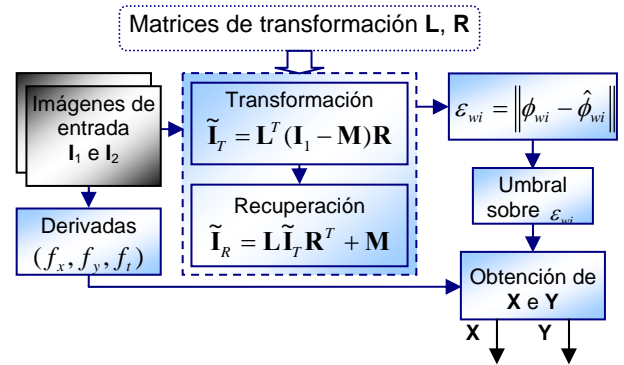


Fig. 4. Diagrama de bloques del proceso de selección de píxeles candidatos

Para la comparación de las nuevas imágenes con el modelo de fondo en primer lugar, se transforma la imagen, utilizando las matrices \mathbf{L} y \mathbf{R} obtenidas anteriormente (10) y a continuación se recupera (11):

$$\mathbf{I}_T = \mathbf{L}^T (\mathbf{I} - \mathbf{M}) \mathbf{R} \quad (10)$$

$$\hat{\mathbf{I}} = \mathbf{L} \mathbf{I}_T \mathbf{R}^T + \mathbf{M} \quad (11)$$

El error de recuperación se define como la diferencia entre la imagen original \mathbf{I}_i y la recuperada \mathbf{I}_{iR} , y puede obtenerse de forma directa restando la imagen recuperada de la original, de forma que el error de recuperación en el píxel de coordenadas (w,i) se define según la ecuación (12) donde \mathbf{I}_{wi} es el valor del píxel en la imagen original e $\hat{\mathbf{I}}_{wi}$ su valor en la imagen recuperada.

$$\varepsilon_{wi} = \left\| \mathbf{I}_{wi} - \hat{\mathbf{I}}_{wi} \right\| \quad (12)$$

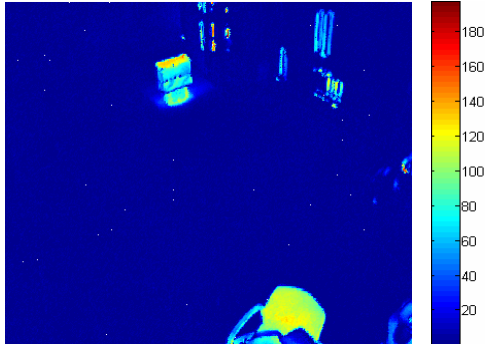
Sin embargo, esto es poco robusto al ruido. Por ello se define una ventana en torno a cada píxel de la imagen y se obtiene el error de recuperación para esa ventana. Si se definen ventanas cuadradas de $q \times q$ píxeles y se identifican por Φ_{wi} y $\hat{\Phi}_{wi}$ para la imagen original y recuperada respectivamente. El error de recuperación asociado al píxel central de la ventana y cuyas coordenadas son (w, i) se determina como:

$$\varepsilon_{wi} = \left\| \Phi_{wi} - \hat{\Phi}_{wi} \right\| \quad (13)$$

El valor del error de recuperación depende fuertemente del número de autovectores utilizados en la generación del modelo de fondo, ya que, a medida que d aumenta, se conservan más componentes principales en la transformación de forma que al perderse menos cantidad de información, el error de transformación también debe disminuir.



(a) Imagen de entrada



(b) Error de recuperación

Fig. 5 Imagen de entrada y error de recuperación.

En la Fig. 5 se puede apreciar que, con las imágenes empleadas en nuestras pruebas, al representar el error de recuperación aparecen varias zonas diferentes. Para comprobar el efecto del número de autovectores en el error de recuperación, se han medido los valores de dicho error tanto en píxeles pertenecientes al fondo, como en píxeles que pertenecen al robot. El resultado se presenta en la Tabla 1, donde se puede apreciar que existe una gran diferencia entre el error de recuperación obtenido en píxeles que pertenecen al fondo de la imagen, y el medido en píxeles sobre el robot, debido a que en estos últimos se ha producido un cambio

importante entre las imágenes utilizadas en la generación del modelo, y las que se quieren segmentar.

Además, se observa que, como cabía esperar, a medida que aumenta el número de autovectores, disminuye el error de recuperación. Este efecto no se da en los puntos del fondo, debido a las características del análisis de componentes principales generalizado (GPCA) ya que es capaz de transformar las imágenes que son similares a las utilizadas en la generación del modelo, con la mínima pérdida de información, incluso utilizando pocos autovectores.

	Autovectores				
	1	5	8	12	15
Fondo	1.977	0.4203	1.3487	1.5085	1.5452
Robot	50.390	43.987	43.204	41.922	40.910

Tabla 1. Error de recuperación obtenido en función del número de autovectores considerados

Los píxeles candidatos a pertenecer a objetos en movimiento serán aquellos en los que el valor de recuperación obtenido supere un determinado umbral, ya que esto indicará que, en esos píxeles, existe una diferencia importante con respecto a las imágenes del fondo utilizadas para la obtención del modelo.

C. Segmentación de objetos en movimiento

Para la segmentación de movimiento se emplea el algoritmo algebraico propuesto en [7], utilizando como datos de entrada únicamente las N medidas $\mathbf{X} = [x_1^j \ x_2^j \ 1]^T_{j=1..N}$ e $\mathbf{Y} = [I_{x1}^j \ I_{x2}^j \ I_t^j]^T_{j=1..N}$ en los píxeles candidatos seleccionados empleando la técnica propuesta en el apartado B.

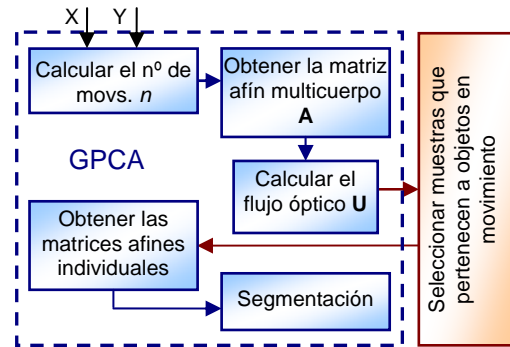


Fig. 6. Etapas del proceso de segmentación de objetos en movimiento

Como se puede observar en la Fig. 6, además de las etapas descritas en [7], se ha añadido un segundo proceso de selección de muestras, empleando como criterio el movimiento de las mismas. Esto se realiza mediante un filtrado sobre las componentes de velocidad en el plano imagen.

IV. RESULTADOS

Para comprobar las prestaciones del algoritmo propuesto se han comparado los resultados que proporciona esta solución con los obtenidos mediante el algoritmo [7]. Todas las pruebas han sido realizadas utilizando un equipo Intel Core 2 CPU 6600 a 2.40GHz con 3.50 Gb de RAM.

En nuestro caso el objeto a segmentar es un robot móvil, y para todas las pruebas que se presentan aquí, se han utilizado 15 imágenes del fondo, y un único autovector para obtener las matrices de transformación

L y R. Los resultados obtenidos al segmentar varias imágenes en las que aparece el robot se presentan en Fig. 7 y Fig. 8. En la primera de ellas se muestra el resultado obtenido al segmentar tres imágenes de una secuencia en las que aparece el robot desplazándose a través de la escena. Se puede observar cómo, al realizar una selección previa de los píxeles, se reduce la detección de falsos puntos, esto es, puntos estáticos que se consideran con el mismo movimiento que el robot (puntos detectados que están fuera del robot), mejorando de esta forma el resultado de la segmentación.

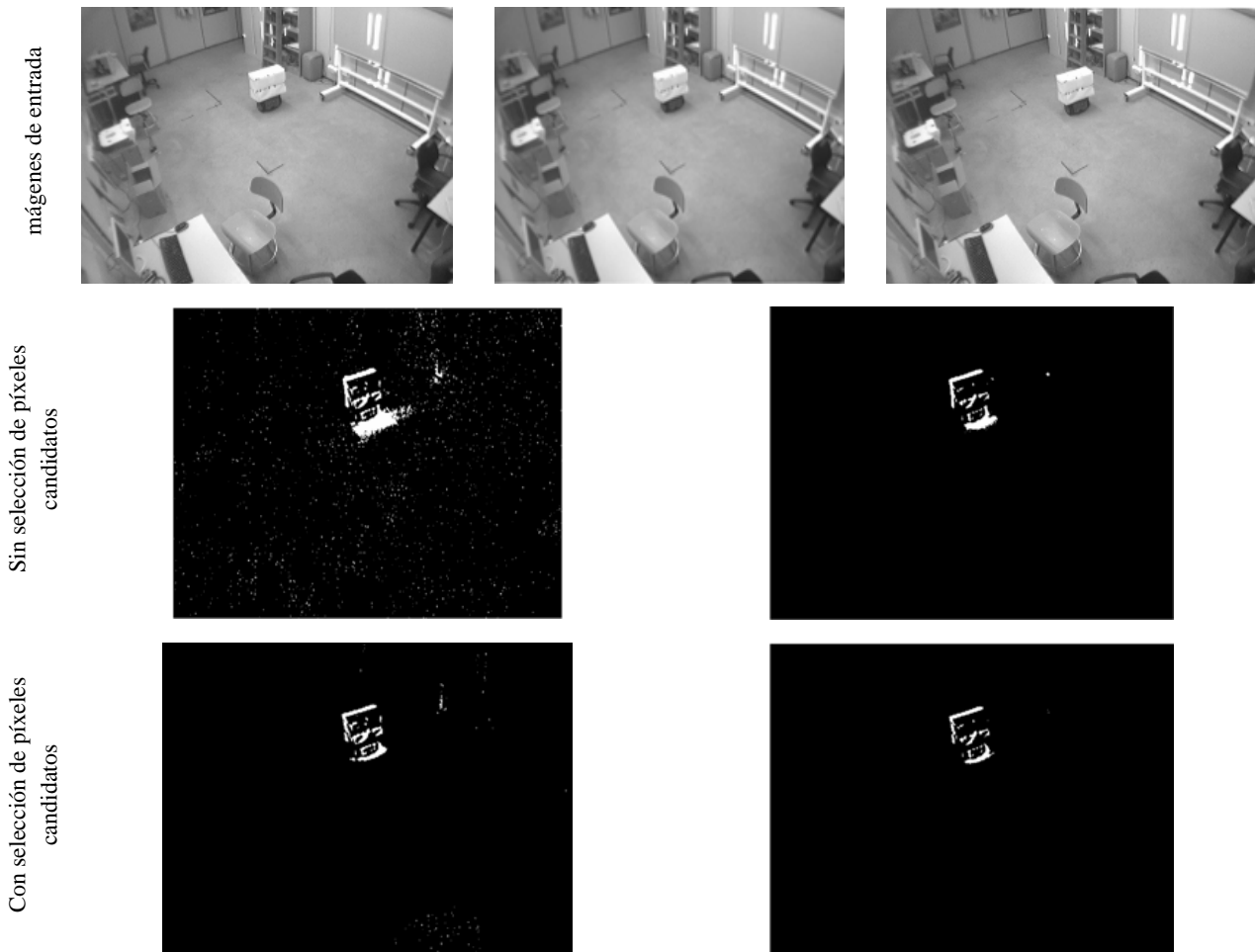


Fig. 7. Ejemplos de resultados de segmentación obtenidos con y sin selección previa de píxeles candidatos

Por otro lado, en la Fig. 8 se ha representado el tiempo de ejecución consumido para la segmentación de diferentes pares de imágenes de entrada. En la imagen superior se presentan los tiempos tanto para los dos algoritmos explicados en este artículo, como para el caso de utilizar el algoritmo de Horn & Schunk [11] para la estimación del flujo óptico, y a continuación la técnica basada en GPCA para la segmentación de movimiento.(Fig. 6) a partir de la etapa de selección de píxeles. Mientras que en la inferior se han representado únicamente los tiempos del algoritmo basado en GPCA

con, y sin la etapa de selección de píxeles para poder apreciar la diferencia entre ambos casos.

En estas figuras también se observa una mejoría al introducir la etapa previa de selección de píxeles, ya que se produce una reducción en el tiempo de ejecución necesario para la obtención del resultado de la segmentación.

A la vista de los resultados anteriores se puede decir que el hecho de introducir una etapa previa a la segmentación, en la que se seleccionan los píxeles que pueden pertenecer a objetos en movimiento presentes en

la escena, provoca una mejora, tanto en la precisión de la segmentación, como en el tiempo de cómputo necesario.

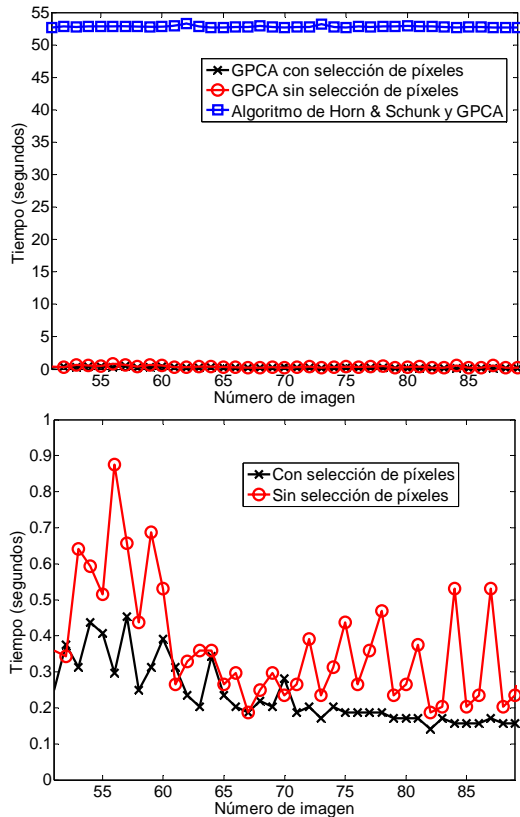


Fig. 8. Tiempos de ejecución del algoritmo de segmentación para cada par de imágenes de entrada

V. CONCLUSIONES

Actualmente existen diferentes alternativas que permiten realizar la segmentación de los modelos de movimiento presentes en una escena, pero todas ellas presentan problemas que no las hacen adecuadas en los casos en que se requiera una segmentación robusta.

La técnica matemática GPCA permite realizar la segmentación de movimiento sin necesidad de conocer de forma previa el número de objetos en movimiento presentes en la escena, sin embargo, su comportamiento se degrada rápidamente en presencia de ruido en las derivadas de las imágenes. Para tratar de disminuir el efecto del ruido, es posible resolver el problema de segmentación aplicando técnicas de minimización no lineal sobre una función de error definida por la restricción afín multicuerpo, pero dada la naturaleza del problema, los resultados obtenidos por este método tampoco son completamente satisfactorios, con la desventaja añadida del aumento en el tiempo de proceso.

En este trabajo se ha propuesto una alternativa que permite la segmentación de uno o varios objetos en movimiento presentes en una secuencia de imágenes

tomadas por una cámara, introduciendo una etapa previa a la segmentación en la que, mediante un técnica de Análisis de Componentes Principales Generalizado (GPCA) se realiza una selección de los píxeles candidatos a pertenecer a objetos en movimiento presentes en la imagen.

En las pruebas realizadas, se ha puesto de manifiesto que el hecho de introducir esta etapa de selección de forma previa a la segmentación mejora notablemente los resultados obtenidos, reduciéndose además el tiempo de proceso necesario para la segmentación.

AGRADECIMIENTOS

Este trabajo ha sido posible gracias a la financiación del Ministerio de Educación y Ciencia (MEC) a través del proyecto RESELAI (REF-TIN2006-14896-C02-01).

REFERENCIAS

- [1] Black, M. and Anandan, P. "Robust dynamic motion estimation over time". Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. June 1991. Page(s): 296-302.
- [2] Yan, H.; Tjahjadi, T.; "Multiple motion segmentation through a highly robust estimator". IEEE International Conference on Systems, Man and Cybernetics Oct. 2004. Page(s): 3082-3087.
- [3] Darrell, T.; Pentland, A. "Robust estimation of a multi-layered motion representation". Proc. of the IEEE Workshop on Visual Motion. 1991. Page(s): 173-178.
- [4] Weiss, Y. "A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models". Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. 1996. Page(s): 321-326
- [5] Costeira, J. and Kanade, T. "A multibody factorization method for independently moving objects". International Journal of Computer Vision. Volume 29, Number 3. September 1998. Page(s):159-179.
- [6] Kanatani, K. "Motion Segmentation by Subspace Separation and Model Selection". Proceedings of the 8th IEEE International Conference on Computer Vision 2001. Volume 2, Page(s): 586-591
- [7] R. Vidal and S. Sastry. "Segmentation of Dynamic Scenes from Image Intensities". IEEE Workshop on Vision and Motion Computing. Page(s): 44-49. 2002.
- [8] Vidal, R.; Yi Ma; Sastry, S.; "Generalized principal component analysis (GPCA)". IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 27, Issue 12, Dec. 2005 Page(s):1945 - 1959
- [9] Turk, M., Pentland, A., "Eigenfaces for recognition", Journal of Cognitive Neuroscience, Vol.3, pp.72-85 (1991).
- [10] Ye, Jieping; Janardan, Ravi; Li, Qi. "GPCA: an efficient dimension reduction scheme for image compression and retrieval". Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining. 2004. Pages: 354 - 363.
- [11] [Horn & Schunck 1981] Horn, B.K.P. and Schunck, B.G., Determining Optical Flow. Artificial Intelligence, N° 17. Page(s):185 - 203. (1981).