# Segmentación de peatones a partir de vistas aéreas

Jorge Ivan Rivalcoba Rivas[1] y Isaac Rudomín[2]

[1] Tecnológico de Monterrey, Campus Estado de México, México
[2] Barcelona Supercomputing Center, Barcelona, España

**Resumen** La simulación de multitudes conducida por datos ha tomado gran importancia en años recientes debido a que permite atribuir comportamientos complejos a las multitudes no logrados por otras técnicas. La información del comportamiento de personas en una multitud es extraída de secuencias de vídeo; para ello es necesario segmentar de manera eficiente a las personas en la escena, los algoritmos desarrollados en el estado del arte están orientados a la segmentación de personas empleando tomas con vistas laterales, las cuales presenta problemas de oclusión, siendo éstas de poca utilidad para el análisis de multitudes. En el presente trabajo se propone un método de segmentación de personas específicamente para vistas aéreas, para ello se emplean como descriptores los Histogramas de gradientes orientados (HOG) en conjunto con un grupo de clasificadores SVM, la novedad de la presente propuesta es la división de las tareas de clasificación condicionadas con la posición de la persona en la escena. Los resultados han permitido la segmentación de personas en vista aérea en tiempo real, la información obtenida de la etapa de segmentación sera usada para alimentar un sistema de seguimiento mismo que generara las trayectorias de los personajes en la escena que serán utilizadas por un simulador de multitudes.

**Keywords:** vista aérea, peatones, segmentación.

## 1. Introducción

La simulación de multitudes por computadora ha progresado de manera significativa desde su concepción ya hace más de dos décadas atrás. En los filmes, videojuegos y en todo tipo de mundos virtuales hemos podido apreciar la simulación de toda clase de multitudes, desde dinosaurios, aves, zombis hasta peatones escapando de desastres naturales. A pesar de que estas multitudes difieren en tamaño y comportamiento, todas en conjunto persiguen un mismo objetivo «*simular una multitud lo mas realista posible a un bajo coste computacional*».

Las multitudes han sido estudiadas desde hace tiempo atrás, como por ejemplo [1] realizo diversos estudios de multitudes durante la revolución francesa, este trabajo fue seguido por [2] el cual centro sus esfuerzos en escenas de pánico,

resultados que fueron aplicados por [3] en teoría de juegos, dichos estudios permitieron deducir que el comportamiento de una multitud no puede ser explicado por el promedio de las acciones de los individuos.

Entre las tareas que más han tenido interés en la simulación de multitudes son:

1. Apariencia.
2. Comportamiento.

La apariencia también llamada «**Rendering**» involucra a todas las técnicas que tienen como objetivo principal ejecutar las tareas de dibujado usando de manera eficiente los recursos computacionales, estas tareas se vuelven primordiales cuando el tamaño de la multitud a simular implica millones de agentes, además de lo anterior, la apariencia también persigue como meta principal que la estética de la multitud provea de la suficiente variedad de personajes permitiendo que esta sea vea más apegada a la realidad.

Por el otro lado, el comportamiento busca que los movimientos de los agentes en la escena se vean lo más plausibles e inteligentes, los trabajos más importantes que atienden la parte cognitiva de los personajes en una simulación suelen caer en cualquiera de estas categorías [4]:

- Autómatas celulares
- Fuerzas sociales
- Sistemas basados en reglas

En cuanto a los sistemas basados en reglas existen los métodos:

1. *Directos*: Basados en reglas.
2. *Indirectos*: Conducidos por datos.

Como ejemplo de simulación basada en reglas se tiene el de [5], el cual enumera un conjunto de reglas simples, que combinadas dotan de comportamientos complejos a los agentes de una multitud virtual, el problema de los sistemas basados en reglas recae en que:

- Requieren un ajuste fino.
- No simulan las variaciones sutiles que se ven en multitudes reales.
- Simulan comportamientos limitados.

Una de las soluciones que se le ha dado a estos problemas es la simulación de multitudes conducida por datos «**Data Driven**», de entre los primeros trabajos que se han presentado donde se utiliza la simulación conducida por datos está el de [6], el cual para obtener los datos que alimentarán la simulación, utiliza una cámara montada en la parte superior de la escena, donde por medio de algoritmos de visión, se realiza el registro del seguimiento de todos los peatones que cruzan un área libre, a partir de esas tomas se generan vectores de posiciones que son agrupadas mediante algoritmos de AI no supervisados, un campo de vectores extrapolado es generado por cada clase o agrupación, todos estos vectores son

usados por un simulador basado en física, cabe destacar que el seguimiento se realiza de manera semiautomática, teniendo la necesidad de la intervención de una persona para la etapa del registro de las trayectorias de cada agente en las tomas de vídeo.

En [7] se presenta otra técnica de simulación de multitudes donde los personajes virtuales exhiben comportamientos que imitan los de humanos reales, para ello se construye un espacio vectorial compuesto de estados $\mathbf{S_i}$ y dinámica de agentes $\mathbf{a_i}$, donde los estados $\mathbf{S_i}$ representan el movimiento de los agentes vecinos, el ambiente, y el movimiento propio del agente, la dinámica $\mathbf{a_i}$ representa un vector bidimensional que corresponde a la velocidad instantánea y la dirección de movimiento del agente en cuestión.

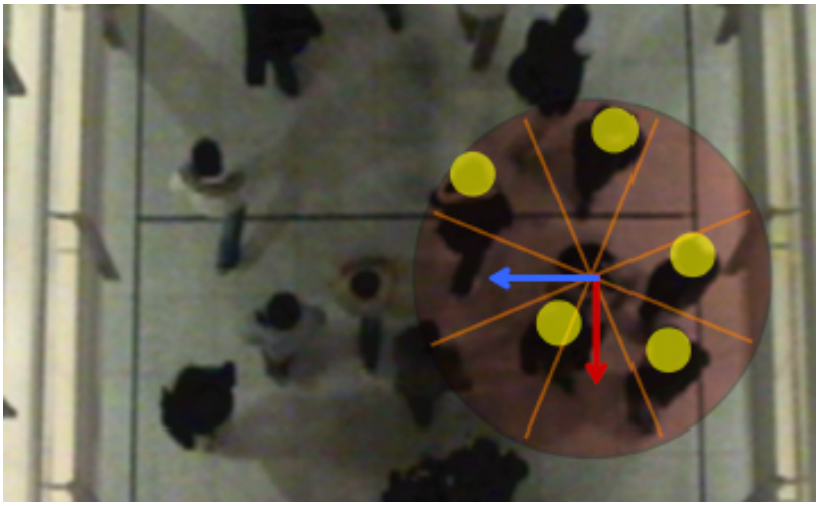

**Figura 1.** En [7] Para lograr capturar cada uno de los estados, se utiliza un circulo radial en el cual se miden la influencia de los vecinos sobre el propietario del círculo.

[8] utiliza algoritmos de visión y define un área de medición alrededor de cada individuo capturado en una escena, con esa información construye una base de datos misma que se usa para buscar el estado más cercano de un agente virtual, en la figura 2 se presenta el proceso mediante el cual se le atribuyen comportamiento a los agentes.

Todos los métodos de simulación previamente presentados y en general los que son conducidos por datos, obtienen la información de comportamiento de secuencias de vídeo, siendo este un proceso arduo que se incrementa proporcionalmente con el numero de personas en la escena.

Para aliviar la fase de recolección de datos los investigadores han empleado algoritmos de visión que asisten la recolección de las trayectorias de cada persona en la escena. En trabajos como el de [9] se realiza la identificación y seguimiento de personas con bastante éxito, sin embargo, debido al ángulo de la toma se

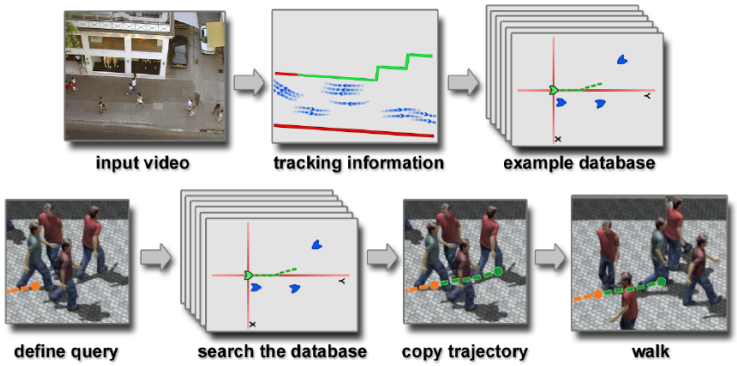


**Figura 2.** Cada una de las etapas presentadas por [8] mediante el cual se logra una simulación conducida por datos a través de buscar en una base de datos la escena mas parecida de acuerdo a las condiciones al rededor del agente que se desea simular.

pierde la información espacial de la posición de las personas en la escena, mismo problema presenta el algoritmo presentado por [10] el cual de manera exitosa logra el conteo de personas en vista lateral pero no recaba ninguna información con respecto a la posición espacial de las personas en la escena, es por esto que las tomas aéreas resultan ser las mas adecuadas en la fase de recolección de datos ya que disminuyen al máximo las oclusiones con respecto a las tomas laterales y aumenta la información en cuanto a las interacciones de las personas en las tomas, si bien los trabajos de reconocimiento y seguimiento de personas en vídeos es un tema bastante trabajado, generalmente no ha sido dirigidos totalmente al análisis de multitudes, existen en el estado del arte trabajos cuya finalidad es la de manejar oclusiones en vistas laterales como en [11] sin embargo el error en la detección de oclusiones suele incrementarse de manera proporcional con el número de personas en la escena, resultando estas técnicas no aptas para ser usadas escenas altamente pobladas. Se han empleado métodos semi-automáticos como el trabajo de [12] donde se emplean marcadores de colores para ayudar al sistema de visión con el seguimiento.

### 1.1. Objetivos

El presente trabajo trata sobre la detección de personas en vistas aéreas con el objetivo de medir las fuerzas sociales entre ellas, todo de manera automática, la información recabada servirá para que en una etapa posterior sirva para un sistema seguimiento mismo que alimentara un sistema de simulación de multitudes que dote a los agentes de comportamiento grupal e individual. Para ello el sistema se divide en dos fases:

1. Detección de los objetos de interés: En esta primer etapa se elimina el fondo de la escena para dejar tan solo los objetos de interés.

2. Detección de personas: En esta segunda etapa se identifican de manera individual a las personas en la escena, posteriormente esta información sera enviada a una etapa de seguimiento y después con la información recabada se creara una base de conocimiento.

## 2. Eliminación de fondo

Esta es una de las etapas mas importantes ya que se encarga de eliminar una gran parte del la información que no es necesaria. Son varios los factores que deben de ser tomados en cuenta a la hora de realizar una eliminación de fondo, [13] presenta una lista de los factores que mas afectan a la eliminación de fondo. El método mas recurrido por su facilidad de implementación es la diferencia entre cuandros contiguis en la captura de vídeo, esto es logrado extrayendo el valor absoluto de la información del cuadro actual con respecto al anterior tal como es expresado en 1. A pesar de ser el método mas preciso presenta el problema de estar condicionado al movimiento, por lo que movimientos demasiado lentos por parte de los objetos de interesen provocaran que sean clasificados como parte del fondo

$$|frame_i| - |frame_i - 1| > Th \tag{1}$$

Un método mas eficiente que tan solo la diferencias de cuadros es el «Running Average» [14] el cual modela el fondo $\mathbf{B_t}$ calculando recursivamente el promedio del valor de cada uno de los pixeles en el escena, cada pixel es clasificado comparando la diferencia entre el cuadro actual $\mathbf{I_t}$ y el modelo del fondo $\mathbf{B_t}$ mediante un nivel de umbral $\mathbf{T}$.

$$D_t = |B_t - I_t| \tag{2}$$

$$M_t(x, y) = \begin{cases} 0, D_t(x, y) \leq T \\ 1, D_t(x, y) > T \end{cases} \tag{3}$$

Una ventaja de este método es la ausencia de correlación espacial entre las diferentes posiciones de los pixeles, es latamente paralelizable siendo muy directa su implementación en GPU sin hembargo una de las desventajas que presenta es la de no ofrecer un método explicito para la selección del umbral $\mathbf{T}$ siendo que este varia de escena a escena. [15] emplea una aproximación de la mediana y la varianza para realizar la clasificación de fondo con contra objetos de interés. En [16] propone un modelo basado en gausianas para realizar extracción de fondo en presencia de sombras, esto es realizado descomponiendo la información de color en sus correspondientes componentes de brillo y cromatismo, asumiendo que la cromaticidad es constante en una sombra pero variable en su brillo. [17] emplea un método similar pero a diferencia de [16], emplea múltiples distribuciones gausianas para describir cada pixel de la escena, estableciendo que las distribuciones con mayor peso y menor varianza perteneces al fondo por el contrario las distribuciones con bajo peso y alta varianza pertenecen a objetos de interés. La información histórica de cada pixel, $\{X_1 ... X_t\}$, es modelada por una

mezcla de $\boldsymbol{k}$ distribuciones gausianas, por lo que la probaiblidad de observación de cada pixel queda determinado por:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{4}$$

Entre las ventajas de modelado por multiples gauianas tenemos el de que es posible tener un modelo de fondo multimodal pero uno de los grandes problemas de este método es la sensibilidad a sobras y cambios de iluminación. además de no especificar como establecer los valores iniciales de esas gausianas. Para sortear los problemas presentados anteriormente y obtener un modelo paralelizable se propone un algoritmo inspirado en el trabajo de [18] modificando la parte de la actualización del fondo la cual queda condicionada al movimiento en la escena siendo que la propuesta original actualiza todo el modelo del fondo resultando esto en un consumo computacional no necesario. El modelado del fondo se realiza por medio de la cuantización y agrupación de las observaciones en un período de tiempo, las observaciones son almacenadas en forma de códigos, al conjunto de códigos se le llama libro de códigos. Los códigos son vectores en el espacio {R,G,B}. La eliminación de fondo BGS(x) para un pixel $\mathbf{x}$ queda definida de la siguiente manera:

1. $x = (R, G, B) I \leftarrow \sqrt{R^2 + G2 + B2}$
2. Para todos los códigos en el libro $\Psi$ encontrar el código $\mathbf{c_i}$ que mas se acerque al valor de $\mathbf{x}$ basandose en las siguientes condiciones:
   - $colordist(\boldsymbol{x}, \boldsymbol{c_m}) \leq \varepsilon_2$
   - $brillo(I, \langle I_{max}, I_{min} \rangle) = \mathbf{true}$
3. El pixel queda determinado como:
   - Objeto: Si no se encuentran el código en el libro
   - Fondo: En caso contrario

En este caso $\varepsilon_2$ es el umbral de detección. La distancia de comparación queda gréficamente expresada mediante la figura 4. Para mantener el fondo se ha determinado utilizar la diferencia de cuadros de modo que tan solo los pixeles estables se actualizarán como fondo, por lo que el sistema de segmentación queda robusto ante cambios de iluminación.

## 3. Detección de personas

El proceso de detección de personas se realiza entrenando un clasificador SVM (se emplea la librería SVM de OpenCV), para alimentar el clasificador se utiliza un descriptor creado a partir de un histograma de gradientes orientadas como el usado por [19], dada una imagen de entrada que es definida por el tamaño de un núcleo de búsqueda, se le realizan las siguientes operaciones:

1. Cada imagen generada por el núcleo de búsqueda es convertida a escala de grises.

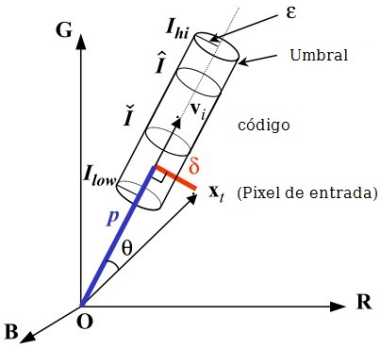


**Figura 3.** La evaluación de la diferencia de un código contra el pixel $\mathbf{x_t}$.

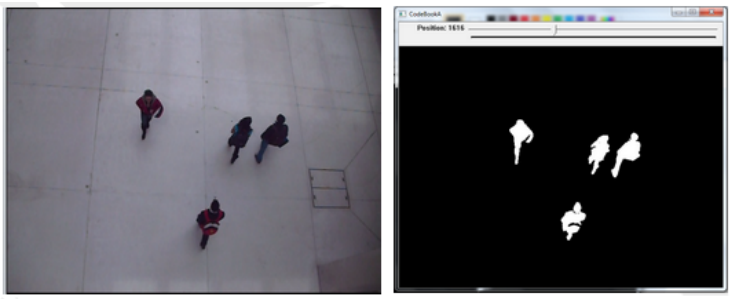

**Figura 4.** Resultados obtenidos con la etapa de extracción de fondo, del lado izquierdo la imagen original, de lado derecho los objetos de interés.

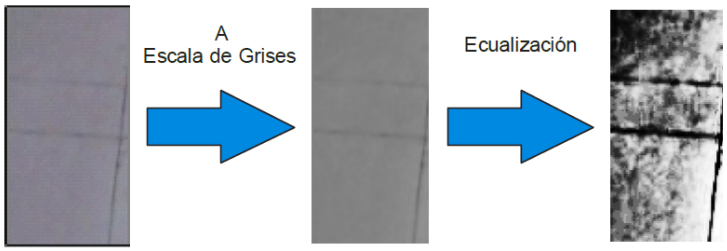


**Figura 5.** Procesamiento realizado al núcleo de búsqueda.

2. Se le realiza una ecualización con la finalidad de robustecer el descriptor ante variantes en la iluminación, este proceso puede observarse en la figura número 5.

Una vez realizado el pre procesamiento, se procede el cálculo de las derivadas direccionadas en «**x**» y «**y**» tal como se muestra en la **fig 6**.


**Figura 6.** Procesamiento realizado al núcleo de búsqueda.

Luego se preparan en memoria arreglos de 9 imágenes vacías que servirán como los bins del histograma, cada una de estas guardara el conteo de las direcciones de los gradientes por cada pixel de la imagen original, cada una guardara un rango de ángulos, además de esto se asigna en memoria otras 9 imágenes la cuales almacenarán la integral de las imágenes anteriores: Por cada pixel de la imagen se calcula el gradiente y la magnitud de dicho gradiente mediante las siguientes ecuaciones:

$$gradiente = atan\left(\frac{SobelY(x,y)}{SobelX(x,y)}\right) * \frac{180}{\pi} + 90 \tag{5}$$

$$magnitud = \sqrt{sobelX(x,y)^2 + sobelY(x,y)^2} \tag{6}$$

Y el lugar donde sera guardada la imagen en el grupo de **bines** estará en función del valor del ángulo del gradiente, por ejemplo si se esta evaluando en $x = 0$ y $y = 0$, y el gradiente da 18° con una magnitud de 30, en este caso se tomara el la imagen cuyos bines correspondan al ángulo de entre 0°-20° guardando en la posición $(0, 0)$ de dicha imagen la cantidad de 30. Así se recorre toda la imagen, produciendo el histograma de ángulos y magnitudes de la imagen. Con esto se genera un vector de características los suficientemente robusto para clasificar la forma humana, este algoritmo presentado en su forma original esta diseñado para detectar personas en secuencias de video, en el presente trabajo realizó una modificación al algoritmo para que permitiera detectar las cabezas para ello:

- Se ajustaron los tamaños de los núcleos de detección a 32 x 32.
- Se realizó una modificación en la generación del **HOG** para obtener histogramas de 0 – 360 grados.

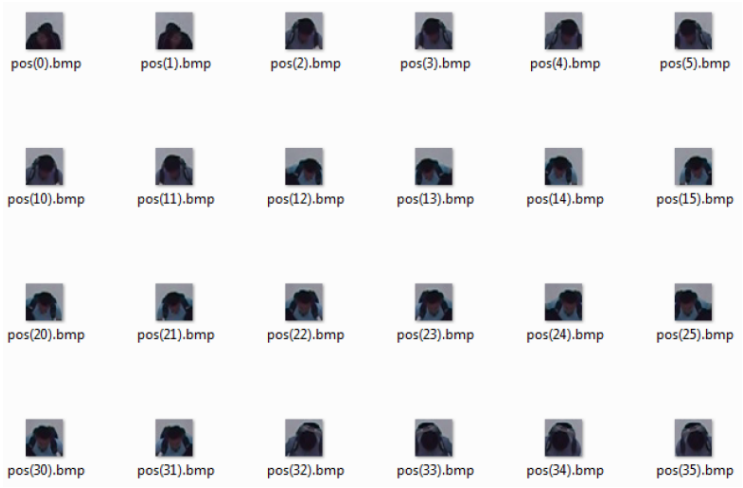


**Figura 7.** Imágenes que muestran el conjunto de datos para entrenar el clasificador SVM con muestras positivas

Se genero así mismo un conjunto de 1000 imágenes de entrenamiento, tanto positivos como negativos figura 7. Una vez aislado el fondo, se segmentan los objetos de interés y por cada uno de ellos se recorre un núcleo de búsqueda que usa el clasificador SVN, permitiendo clasificar cada una de las muestras en cabezas o no cabezas. Se ha determinado seleccionar a las cabezas como el elemento a seguir debido a que permanece inmóvil a lo largo del seguimiento. La granularidad de búsqueda del núcleo clasificador se muestra en la figura numero 9.

## 4. Seguimiento

Para el seguimiento se ha utilizado el método de Lucas-Kanade que integra OpenCV. El algoritmo de seguimiento Lucas-Kanade [20] fue originalmente propuesto en el año de 1981, este método utiliza información local derivada de una ventana de acción sobre el punto de interés, siendo este el principal diferenciador con respecto a otros métodos de seguimiento, las desventajas que presenta este algoritmo de seguimiento es que debido al uso de información local, no da buenos resultados cuando los movimientos en el objeto de interés llegan a ser demasiado amplios, para mejorar este aspecto se procedió a desarrollar una implementación piramidal del método, permitiendo empezar con ventanas sobre una versión sub muestreada de la imagen hasta llegar a la imagen original. Otro aspecto importante en la detección de la cabeza fue contemplar la información de la deformación de la figura humana la cual se incrementa en la medida en que el objeto de interés se aleja de la cámara figura 10. Para ello se realizo una entrenamiento generando diferentes clasificadores los cuales entrarían de acuerdo al mapa de calibración de la escena mostrado en la figura 11.

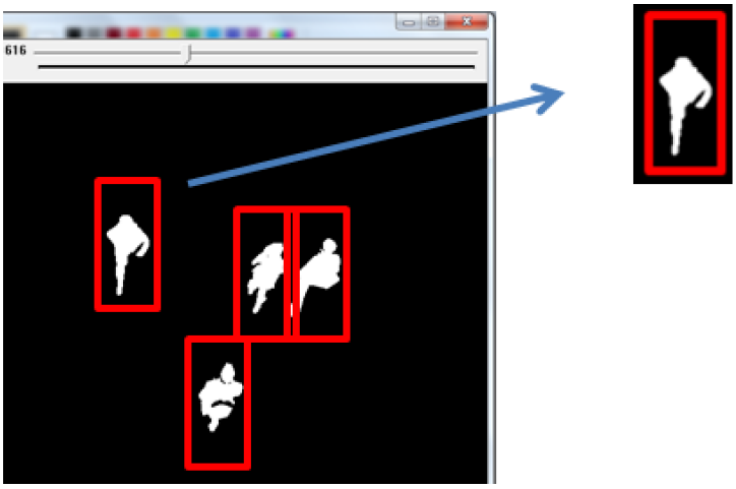


**Figura 8.** Se realiza una detección de blobs y sobre cada uno de ellos se recorre un núcleo de detección de cabezas para poder realizar el seguimiento

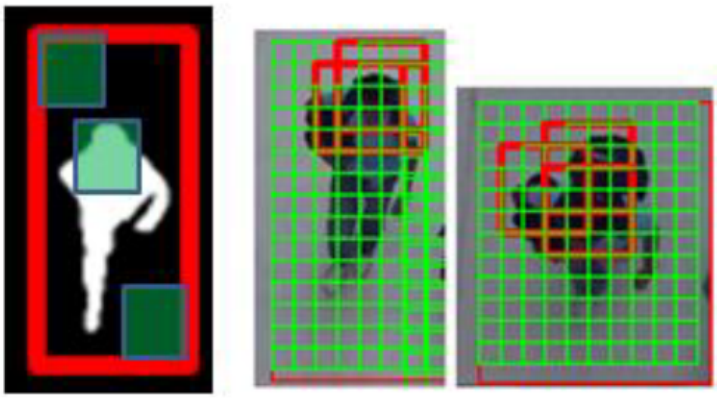

**Figura 9.** En esta imagen se muestra la granularidad del núcleo de detección, este recorrido se realiza por cada blob identificado en la etapa de segmentación

**Figura 10.** En esta imagen se muestra el grado de deformación que presenta una persona cuando se incrementa su distancia con respecto a la posición de la cámara

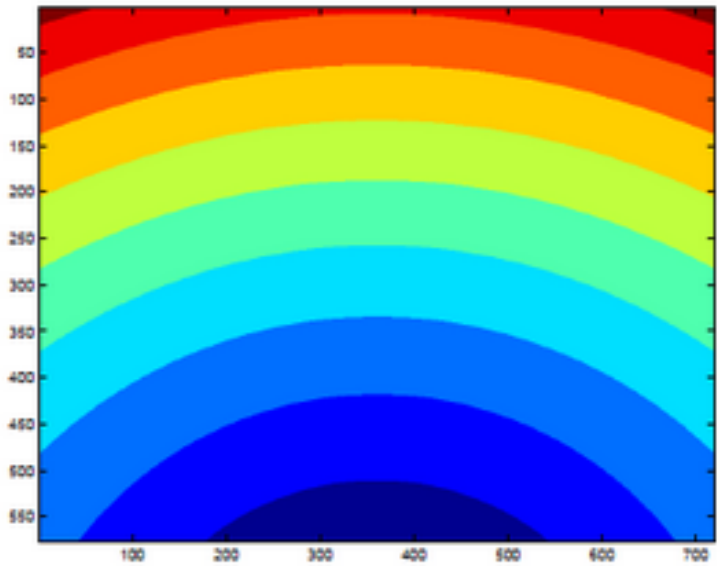


**Figura 11.** En esta imagen se muestra el mapa que condiciona los núcleos de detección, para cada segmento de color se aplica un núcleo de detección diferente.

## 5. Conclusión

Se ha presentado una metodología que permite la segmentación de personas en vista aérea, con el fin de utilizar dicha información para obtener la interacción social de un grupo de personas en una secuencia de vídeo, si bien hay una gran variedad de algoritmos que garantizan la correcta segmentación de personas, estos están diseñados principalmente para vistas laterales, las tomas laterales no son de gran utilidad para el análisis de multitudes ya que reducen en gran porcentaje la información espacial de cada uno de los individuos en la escena. Para la segmentación de personas se han utilizado dos etapas,la primera de ellas elimina el fondo de la escena y deja los objetos de interés, que en este caso son las personas, seguido de eso a cada objetos de interés se le aplica un núcleo de reconocimiento cuya finalidad es detectar las personas en la escena, se emplearon como descriptores los histogramas de gradientes orientados los cuales han probado ser efectivos a la hora de segmentar la figura humana. En trabajo futuro se pretende paralelizar cada una de las fases del sistema aquí presentado para incrementar la velocidad a la cual se obtienen los resultados.

## Referencias

1. Le Bon, G.: The Crowd Study of Popular Mind by Gustave Le Bon. CreateSpace Independent Publishing Platform (2013)

2. Michalewicz, Z., Fogel, D.B.: How to Solve It: Modern Heuristics (2000)
3. Teitelbaum, P., Epstein, A.: This Week's Citation Classic. Psychol. Rev (7) (1962) 1981
4. Charalambous, P., Chrysanthou, Y.: Learning Crowd Steerin Behaviors from Examples. In And, R.B., And, Y.C., Komura, T., eds.: Motion in Games - Third International Conference. Volume 6459., Springer (2010) 35
5. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. ACM SIGGRAPH Computer Graphics **21**(4) (August 1987) 25–34
6. Musse, S.R., Jung, C.R., Jacques, J.C.S., Braun, A.: Using computer vision to simulate the motion of virtual agents. Computer Animation and Virtual Worlds **18**(2) (May 2007) 83–93
7. Lee, K., Choi, M., Hong, Q., Lee, J.: Group behavior from video: a data-driven approach to crowd simulation. In: Proceedings of the 2007 ACM . . . , San Diego, California (2007) 109–118
8. Lerner, A.: Crowds by example. Computer Graphics Forum **26**(3) (September 2007) 655–664
9. Seemann, E., Leibe, B., Schiele, B.: Multi-aspect detection of articulated objects. Computer Vision and Pattern . . . **2** (2006) 1582–1588
10. Subburaman, V.B., Descamps, A., Carincotte, C.: Counting People in the Crowd Using a Generic Head Detector. In: 2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance, IEEE (September 2012) 470–475
11. Nevatia, R.: Segmentation and tracking of multiple humans in complex situations. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. Volume 2., IEEE Comput. Soc (2001) II–194–II–201
12. Xiaoyu Deng Coll. of Comput. Sci., Zhejiang Univ., H., Liu, J.B.Z.Y.C.C.Y.: A block-based background model for video surveillance. In: 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE (March 2008) 1013–1016
13. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: principles and practice of background maintenance. Proceedings of the Seventh IEEE International Conference on Computer Vision (1999) 255–261 vol.1
14. Mayo, Z., Tapamo, J.: Background subtraction survey for highway surveillance. Proc. of the Annual Symposium of the Pattern . . . (2009)
15. Manzanera, A., Richefeu, J.C.: A new motion detection algorithm based on $\Sigma$–$\Delta$ background estimation. Pattern Recognition Letters **28**(3) (February 2007) 320–328
16. Porikli, F., Tuzel, O.: Bayesian background modeling for foreground detection. Proceedings of the third ACM international workshop . . . (2005) 55
17. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149) 246–252
18. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground–background segmentation using codebook model. Real-Time Imaging **11**(3) (June 2005) 172–185
19. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) **1** (2005) 886–893
20. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. Proceedings of the 7th international joint . . . **130** (1981) 121–130