

# Acoplamiento de micro multitudes para el desarrollo de videojuegos controlados por movimiento

Iván Rivalcoba<sup>1</sup>, Krely Rodríguez<sup>2</sup>, Oriam Degives<sup>1</sup>, Isaac Rudomín<sup>3</sup>

<sup>1</sup> Tecnológico de Monterrey, Campus Estado de México,  
México

<sup>2</sup> Tecnológico de Minatitlán,  
Minatitlán, Veracruz, México

<sup>3</sup> Barcelona Supercomputing Center  
Barcelona, España

**Resumen.** La simulación de multitudes en tiempo real y los juegos controlados por movimiento se han vuelto muy populares en los últimos años. En conjunto estas dos tecnologías proporcionan una mejor experiencia de juego en entornos virtuales logrando escenas más realistas y vibrantes. Sin embargo, hasta ahora no se ha explotado la interacción de múltiples jugadores con una gran multitud bajo un entorno virtual. En este trabajo presentamos un sistema no intrusivo capaz de simular multitudes virtuales acopladas en tiempo real con varios usuarios, sentando con ello las bases para la creación de juegos donde interactúen muchos jugadores con muchas personajes, para ello se realiza una detección de personas en una secuencia de vídeo, nuestra contribución consiste en utilizar patrones binarios semánticos locales (S-LBP) como descriptores, creando con ello un sistema inteligente capaz de segmentar detectar y seguir personas en vistas aéreas, así mismo para aumentar la velocidad del dibujado y acoplamiento se aprovecha la unidad de procesamiento gráfico (GPU), almacenando la información en *texture shaders* delineando una arquitectura de procesamiento concurrente. Los resultados experimentales muestran un sistema viable para la interacción de multitudes virtuales contra más de una docena de personas reales simulados en hardware de consumo y en tiempo real (30 FPS), así mismo el uso de los patrones binarios semánticos locales como descriptores permitieron reducir los falsos positivos que normalmente se producen al emplear descriptores HOG (Histograma de gradientes orientados) usados frecuentemente en el reconocimiento de personas.

**Palabras clave:** Micro multitudes, videojuegos controlados por movimiento.

## 1. Introducción

La simulación de multitudes en tiempo real se ha convertido en una de las principales líneas de investigación en los juegos de computadora, películas y realidad virtual [27]. Sin embargo, solo unos pocos juegos utilizan las multitudes

como una forma de mejorar la experiencia de juego, esto se debe en gran medida a la capacidad limitada en el hardware de los usuarios finales, especialmente de las consolas de juegos. Los vídeo juegos han utilizado la simulación de multitudes para lograr escenas más realistas y vibrantes [8, 23].

Películas como la trilogía del Señor de los Anillos, Avatar y Guerra Mundial Z, hacen un amplio uso de las técnicas desarrolladas para la simulación de multitudes. Miles de personajes complejos interactúan entre sí y con su ambiente, alcanzando un sentido de inmersión más profundo en los espectadores. Sin embargo, debido a la naturaleza de la industria cinematográfica, en la cual se da prioridad al realismo, el empleo de simulación fuera de línea es la opción típica, por lo que la exigencia en el tiempo no es crucial; por otro lado los videojuegos son un caso aparte, debido a que el tiempo de dibujado por cada cuadro es crucial para no experimentar retardos en la animación, por lo que esta industria requiere de algoritmos más optimizados para lograr escenas en milésimas de segundo [7].

Los juegos controlados por movimiento se han vuelto muy populares en los últimos años con la introducción del hardware dedicado a la detección de movimiento, como el *Kinect* de Microsoft y el *Playstation Eye* de Sony. Sin embargo, en la medida de nuestro conocimiento, no existe juego que combine ambas características, el permitir a varios usuarios interactuar con una gran multitud virtual. El objetivo principal del presente trabajo es llevar a cabo el acoplamiento entre un grupo de personas reales y la simulación de una multitud de humanos virtuales. Para ello, proponemos un sistema interactivo de dos etapas. En la primera etapa un sistema de visión detecta personas desde una secuencia de vídeo todo ello en tiempo real. Esta etapa no requiere de una cámara especial como los otros sistemas de detección mencionados, una cámara web de hasta 3MB de resolución ofrecerá la calidad necesaria para permitir el funcionamiento óptimo del sistema. La posición de los usuarios capturados es utilizada en la segunda etapa, dicha etapa consiste en realizar el acoplamiento entre multitudes virtuales y usuarios. Con ello se logra que los personajes virtuales reaccionen ante la presencia de los humanos reales.

### 1.1. Trabajo relacionado

Una parte crucial del sistema interactivo presentado en este artículo, es la cantidad de tiempo requerido para realizar las tareas de simulación y presentación. Para obtener una comprensión más profunda del sistema, esta sección describe los desafíos y el estado del arte relacionado con la visión artificial y la simulación de multitudes.

Thalmann et al. presenta un sistema interactivo de navegación en el cual el usuario tiene la capacidad de controlar un personaje en la multitud a través de una interface natural, pero la interacción sigue siendo entre un único usuario y personajes virtuales [27]. Pelechano propuso un experimento para estudiar de cerca el comportamiento de las personas, en una fiesta de cocktail, interactuando con personajes virtuales [21].

Helbing et al. presentó el modelo de fuerzas sociales, en el cual, cada agente interactúa con el medio ambiente dependiendo de sus motivaciones internas [10]. Helbing impulsa su investigación y estudios, en el comportamiento de los peatones en condiciones normales y en situaciones de evacuación [9]. El modelo de las fuerzas sociales se ha ampliado para apoyar a los grupos de peatones relacionados entre sí [16].

Treuille et al. presentó el modelo de Multitudes *Continuum*, el cual unifica las etapas de planificación y ruta de evasión de colisión mediante el uso de la velocidad y campos potenciales. [24]. Millan et al. utilizó las fuerzas de atracción y repulsión codificadas en texturas para dirigir los agentes a través del medio ambiente [15]. Van den Berg et al. Propuso *Reciprocal Velocity Obstacles*, un conjunto de velocidades que conducirán a una colisión entre agentes [4]. Los agentes deben elegir velocidades que no pertenezcan a este conjunto para moverse a través de rutas libres de colisiones. Los mismos autores redujeron el problema a la solución de un sistema lineal de baja dimensión [3]. Un enfoque de prevención de colisiones basado en visión es presentada por Ondřej et al., usando visión sintética para detectar obstáculos en el campo de vista de cada agente [19].

En cuanto al reconocimiento de humanos, existen varias obras en la comunidad de investigación que se ocupan de este problema. Viola et al. Presento una técnica para la detección de peatones en una secuencia de vídeo, mediante el aprovechamiento de la apariencia del caminar de una persona [26]. Dalal et al. realizó un estudio completo de los Histogramas de gradientes orientados (*HOG*) aplicado a la representación de humanos [6] para detectar peatones con éxito. Tuzel et al. detectó peatones mediante la representación de una región de la imagen como matrices de covarianza de localidades espaciales, intensidades, y derivadas, por nombrar algunas [25].

En la mayoría de los casos, los sistemas de detección incluyen una etapa de preprocesamiento, la cual consiste en la eliminación del fondo de la escena. Banerjee et al. Convino un modelo de fondo adaptable con características descriptores *HOG*, aprendiendo solo las partes fijas de la escena e ignorando el primer plano en movimiento [2]. Otro enfoque es presentado por Bhuvaneshwari et al. Llamados rasgos *edgelet*, los cuales son segmentos cortos que pertenecen a una línea o curva [5].

Los trabajos mencionados anteriormente se ocupan de escenas en perspectiva lateral, las cuales son deseables para aplicaciones de video vigilancia, sin embargo, no son recomendados cuando se trata con el análisis de multitudes, debido a la oclusión que se presenta cuando el tamaño de la multitud se incrementa, resultando en una decaimiento de la precisión en los sistemas de reconocimiento. Lengvenis et al. Utilizó una perspectiva con vista aérea (ojo de pájaro) para la detección de los peatones en el transporte público de la ciudad de Kaunas [13]. Ozturk et al. Propuso un sistema para determinar la orientación de la cabeza y cuerpo de los humanos en las escenas de visión aérea [20]. Su sistema utiliza una aproximación de contextos de forma para detectar la orientación básica del cuerpo y propone un flujo óptico basado en las propiedades de la transformación característica invariante a la escala (*SIFT*), sin embargo, las oclusiones se dejan

como trabajo futuro, haciendo a este algoritmo inutilizable para escenas con multitudes.

## 2. Arquitectura del sistema

El sistema que describimos en este trabajo está diseñado para ser interactivo incluso en un hardware de consumo, siendo capaz al mismo tiempo de capturar peatones desde las imágenes de vídeo, simulando miles de agentes virtuales. Hemos implementados las técnicas descritas usando C++ para el código en CPU, y OpenGL moderno para la simulación y visualización, usando un sistema operativo Windows 7 x64 con un procesador Intel Core i7-2630 QM de cuatro núcleos a 2.00 GHz, 8GB de memoria y tarjeta gráfica NVIDIA GT 540M. El GT 540M es una tarjeta gráfica de la familia Fermi con la capacidad para manejar la versión de OpenGL 4.3; es considerada una GPU de gama baja-media. Para las etapas de dibujado y simulación hemos empleado el API de OpenGL moderno específicamente en su versión 4.3.

OpenGL introdujo los *Computer Shaders* a su API, el cual habilita su lenguaje de *shaders* para llevar a cabo cómputo de propósito general en hardware gráfico. Los *Compute Shaders* proporcionan un enfoque alternativo al cómputo de propósito general en unidades de procesamiento gráfico (GPGPU), además de Nvidia CUDA y OpenCL del Grupo Khronos. OpenGL es más accesible que Cuda, ya que a diferencia de Cuda y OpenCL, los *Compute Shaders* son similares a otras etapas de sombreado en la pipeline de gráficos y son capaces de leer desde texturas, imágenes y buffers así como escribir imágenes y buffers sin mapeo a otras estructuras de datos o cambiar entre los gráficos y contexto de cálculo. Estas características proporcionan una clara ventaja para las aplicaciones que requieren interoperabilidad de gráficos, como los videojuegos. Una visión general del sistema puede observarse en la figura 1.

Para lidiar con los problemas de oclusión presentados en escenas con una gran cantidad de personas, se decidió utilizar una perspectiva aérea también conocida como vista de pájaro. Por otra parte también se tuvo cuidado en los procesos que involucraban la segmentación y eliminación de fondo, de modo que el sistema pudiera enfocar sus esfuerzos computacionales en las tareas de clasificación, así mismo se investigó la mejor forma de representar humanos para las etapas de reconocimiento de personas en filmaciones desde vistas aéreas, para eso se consideró la deformación que se presenta cuando un humano se aleja de la cámara, para lo cual se tomó la decisión de segmentar la acción de los clasificadores, haciendo dependiente el tipo de clasificador empleado con respecto a la distancia entre el usuario y la cámara.

El sistema que realiza la detección humana consiste en 6 etapas mostradas en la Figura 2. La primera etapa consiste en la captura y digitalización del vídeo, posteriormente se realiza la sustracción del fondo, esta etapa es responsable de la división de la escena en el fondo de las regiones de primer plano. Las regiones de primer plano son etiquetadas como Regiones de interés (ROI), la separación

permite al sistema concentrar los esfuerzos computacionales en las tareas de detección de humanos (objetos en movimiento).

La sustracción del fondo se lleva a cabo utilizando una versión modificada del algoritmo codebook presentado por [12]. Para hacer el algoritmo más robusto contra los cambios de iluminación, empleamos un segundo codebook que es guardado en cada cuadro; este codebook auxiliar genera unas palabras claves considerando solo las regiones sin movimiento. Una vez que ha transcurrido cierto lapso de tiempo, este codebook es limpiado y es adherido a un codebook principal. Una vez removido el fondo, las ROIs, son etiquetadas y convertidas de *Red-Green-Blue* (RGB) a una escala de grises.

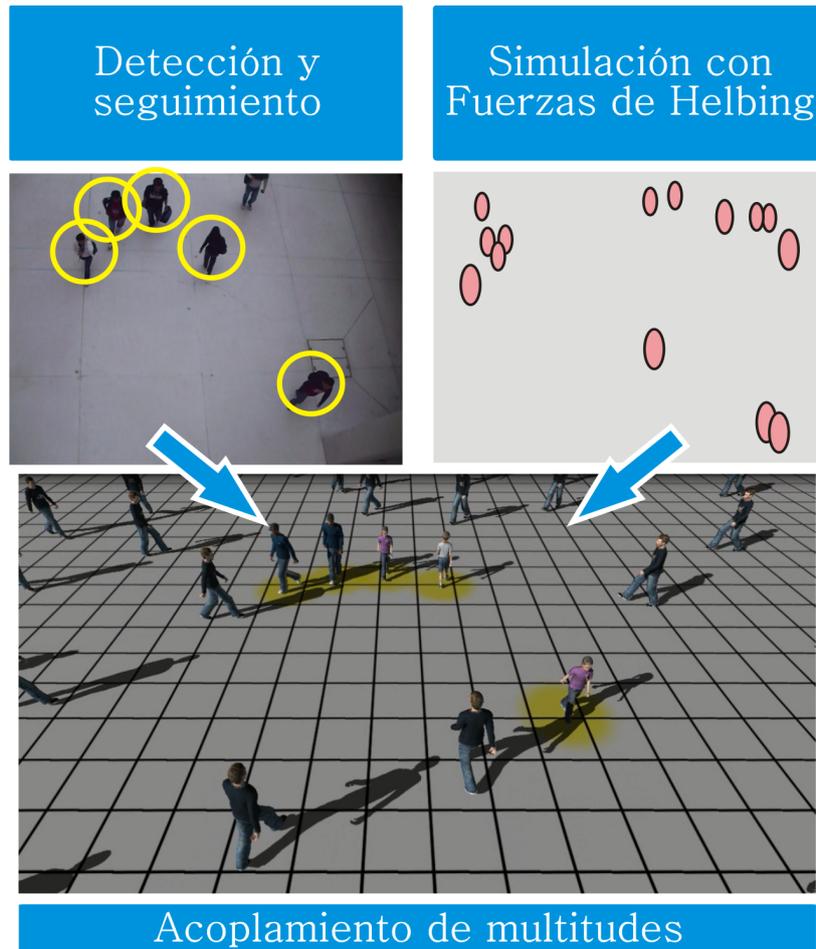
Una vez que se han obtenido el conjunto de regiones ROI, en una tercera etapa las ROI son pasadas una a una al módulo de extracción de características, en este caso los descriptores seleccionados para modelar la parte superior del cuerpo humano son una versión modificada de patrones binarios locales (LBP) presentados por [18], estos son llamados patrones binarios locales semánticos, utilizando estos descriptores se genera un vector de características que modela el objeto detectado como un histograma, Cada uno de estos vectores es enviado a una máquina de vectores de soporte (*SVM*) entrenada previamente misma que realiza la clasificación etiquetando la ROI como humano o no humano. Posteriormente en la cuarta etapa se realiza el seguimiento de cada personaje utilizando el algoritmo de *Lucas-Kanade*, cada uno de estos puntos es almacenado en la quinta etapa en *texture shaders* siendo enviados a la *GPU* para ser dibujados con los personajes virtuales.

Se tomó la decisión de seguir el área correspondiente a la cabeza y hombros del cuerpo humano debido al hecho de que la cabeza humana permanece casi constante sobre toda la escena [17], a diferencia de otras partes del cuerpo. La sección de cabeza y hombros presentan una forma característica en forma de  $\Omega$ , ver figura 3. Esta propiedad es la que hace que la cabeza y hombros sean las partes más estables del cuerpo para ser detectadas y rastreadas.

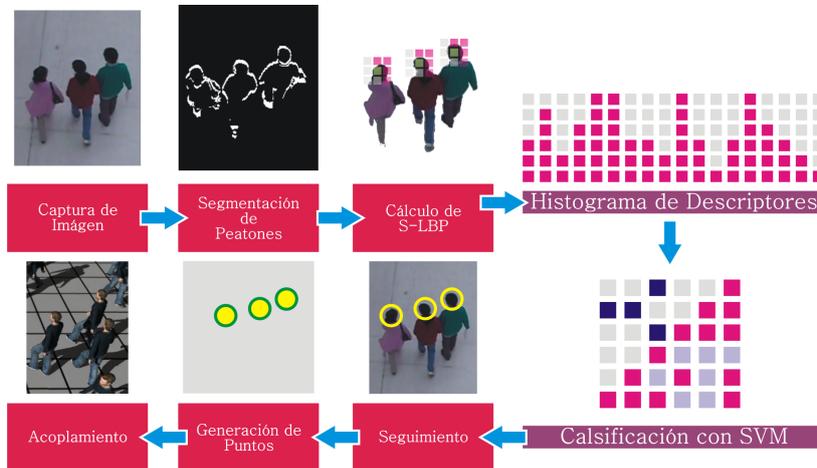
## 2.1. Sistema de reconocimiento usando *S-LBP*

Nuestra principal aportación recae en la etapa de detección de personas la cuál es implementada de la siguiente forma: Un operador LBP asigna una etiqueta a cada píxel de una imagen por un umbral a la zona de 3x3 de cada píxel con su propio valor y considerando el resultado como un número binario, por ejemplo dado un píxel  $(x_c, y_c)$ , el LBP de ese píxel es calculado usando la ecuación 1, donde  $P$  corresponde al número de puntos de muestreo,  $R$  es el radio de la zona a muestrear  $g_p$  representa cada punto de muestreo y  $g_c$  representa el píxel  $(x_c, y_c)$ ; el histograma de los números obtenidos a lo largo de toda la imagen es usada como el descriptor.

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (1)$$



**Fig. 1. Flujo del sistema.** Los datos de los peatones capturados por la etapa de detección se fusionan con la simulación virtual produciendo un acoplamiento entre la parte virtual y real, con esto se consigue que la animación en el ambiente virtual sea influenciada por los peatones capturados en la secuencia de vídeo



**Fig. 2. Etapas del sistema de acoplamiento.** Se muestra cada una de las seis etapas que componen el sistema de acoplamiento

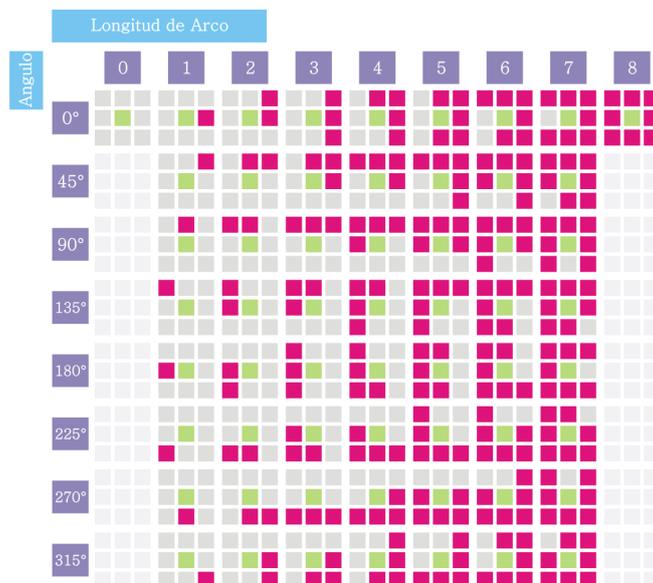


### Forma Omega Cabeza-Hombros

**Fig. 3. Forma omega del cuerpo humano.** Desde una vista aérea un ser humano muestra una silueta tipo omega característica, muy útil en los procesos de reconocimiento y segmentación, la silueta se preserva a pesar de los cambios sutiles de perspectiva

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & otherwise \end{cases} \quad (2)$$

Aunque el operador LBP tiene muchas ventajas sobre otros descriptores basados en gradiente, la LBP no es suficientemente robusta para ser usada como descriptor humano, sin embargo sus puntos fuertes pueden ser explotados usando una variante llamada *LBP semántica* (S-LBP) propuesta por [11]. S-LBP está formado de LBP uniformes. Un LBP se llama uniforme si se presentan como máximo dos transiciones de bit a bit entre 0 y 1, en otras palabras, varios bits continuos “1” forman un arco en el círculo de muestreo; estos arcos puede ser representados con su dirección principal y la longitud del arco (ver figura 4); mediante el uso de esta estrategia, la dimensión del vector característico es reducido.



**Fig. 4. Patrones Locales Binarios Semánticos.** La figura muestra las 58 combinaciones semánticas de los LBP (Patrones Locales binarios - Local Binary Patterns), cada columna representa a la longitud del arco, por otra parte las filas representan el ángulo principal del LBP

En nuestro trabajo, utilizamos un núcleo de detección de 32x32 pixeles de tamaño, el cual produce un vector 1160D; todos los S-LBPs son usados para calcular un histograma, cada histograma representa el vector característico de cada región. Para acelerar el proceso de cálculo de SLBPs, proponemos el uso de una tabla de búsqueda para la clasificar el tipo de S-LBP. Se ha demostrado por

[25] que el conjunto de sub-ventanas de tamaño variable puede favorecer en gran medida la eficiencia de detección. Debido a esto, hemos adoptado un enfoque de varias resoluciones similares a [1], dividiendo cada núcleo de detección en ventanas locales de diferentes tamaños para capturar mejor la estructura principal de la cabeza y hombro de los seres humanos. El histograma de descriptores S-LBPs es ahora extraído para cada ventana de forma independiente (formando un total de 20 subventanas) y luego concatenada para formar una descripción global de la región. Esta estrategia de extracción de características es utilizada tanto para entrenar el sistema y para clasificar la imagen en tiempo de ejecución. Utilizamos la librería OpenCV SVM para la generación del vector de apoyo para la tarea de clasificación, el conjunto de datos utilizado es similar [14], al que utiliza la vista superior de tiros de cabeza y los hombros humanos. Una observación importante es que los peatones muestran una deformación debido a la perspectiva de la cámara, haciendo difícil la sintonización hasta el clasificador [22].

Para superar este problema, y así mejorar los resultados de la detección, dividimos la vista de la cámara en segmentos en función de la distancia entre la cámara y el peatón; un clasificador diferente es usado dependiendo de la región donde el usuario es detectado. Una vez que la cabeza humana es detectada, un punto es asociada a ella; cada punto detectado es rastreado sobre todos los marcos usando la versión piramidal de rastreo de Lucas Kanade. Para nuestro sistema de simulación, toda lógica relacionada con el control del agente se realiza en el hardware de gráficos, el cual nos permite realizar simulaciones a gran escala con miles de agentes en tiempo real. Para la interoperabilidad entre la representación y la simulación, almacenamos todos los datos requeridos para la simulación en la memoria de textura. Los datos registrados por el sistema de visión son almacenados en las mismas texturas que los utilizados por los agentes simulados, y la representación visual del arreglo entre los peatones y los agentes simulados puede observarse en la Figura 5.

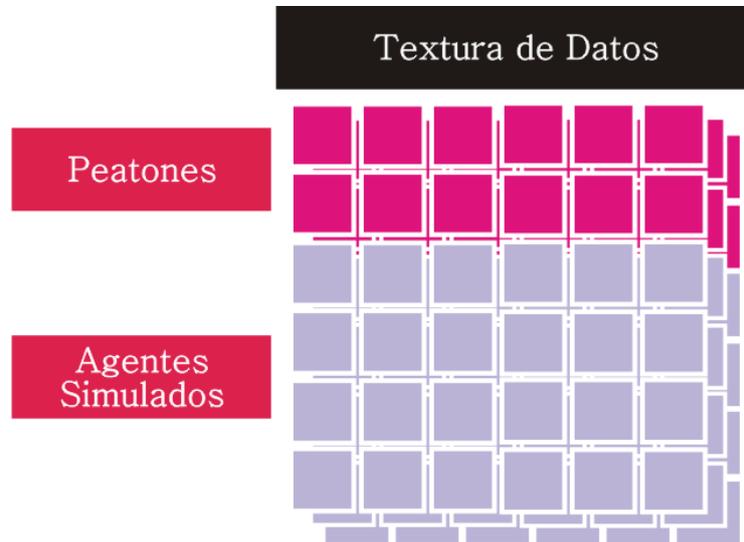
## 2.2. Comportamiento de los agentes virtuales

Para dirigir los agentes simulados, utilizamos una implementación paralela del trabajo realizado por Moussaid et al [16]. Se requieren tres texturas de datos para la simulación: texturas de posición, velocidad y destino.

Cada elemento de textura (*texel*)  $i$  contiene los datos del agente o usuario  $i$ . Nosotros utilizamos la ecuación (3) para calcular la nueva posición de cada agente simulado:

$$\mathbf{f}_i = \mathbf{f}_i^d + \mathbf{f}_i^n + \mathbf{f}_i^o + \mathbf{f}_i^g \quad (3)$$

Donde  $\mathbf{f}_i$  representa la aceleración del agente  $i$  en el cuadro actual, y es afectado por motivaciones internas  $\mathbf{f}_i^d$ , sus vecinos  $\mathbf{f}_i^n$ , los obstáculos más próximos  $\mathbf{f}_i^o$ , y su grupo dinámico  $\mathbf{f}_i^g$ . La nueva velocidad  $\mathbf{v}_{i_{new}}$  se obtiene mediante la adición de la aceleración a la velocidad anterior del agente  $\mathbf{v}_i$ . Finalmente, la posición del agente es actualizada en sus componentes de aceleración y velocidad.



**Fig. 5. Almacenamiento de la información en texturas.** Los datos de los peatones capturados en las secuencias de vídeo son almacenadas en las mismas texturas que los agentes simulados, permitiendo un cómputo más eficiente así mismo se aprovechan las capacidades de cómputo de propósito general de las unidades de procesamiento gráfico

Detectamos vecinos basados en sus posiciones mediante un algoritmo *hashlike* espacial. El algoritmo busca los vecinos más cercanos utilizando las posiciones de textura. Debido a que tanto los agentes simulados y los peatones capturados almacenan sus posiciones en esta textura, los agentes simulados son conscientes de los peatones reales capturados, justo como si fueran otros agentes virtuales. Al ser conscientes de los peatones capturados, los agentes virtuales pueden reproducir tareas de evasión y realizar cualquier otra forma de interacción de la misma manera que con los agentes virtuales vecinos. Solo existen unas pocas obras en el estado del arte que tienen que ver con la interacción entre un grupo de personas reales y una multitud virtual, sin embargo, estos sistemas son típicamente muy limitados en el número de usuarios capturados o en el tamaño de la multitud. Por otro lado, la segmentación de peatones se realiza generalmente utilizando procesos manuales o procesos semi-automáticos, o en formas más invasivas como confiar en el uso de marcadores.

### 3. Resultados

Las vistas frontales dedicadas al uso de hardware, como el Kinect y PlayStation Eye, limitan el área y el número de personas que pueden ser detectados. Los resultados muestran que nuestro sistema puede realizar una interacción en tiempo real entre un grupo de usuarios y un grupo simulado de personajes virtuales. Con el uso de un codebook auxiliar para el modelo de fondo, el sistema

es capaz de mantener su funcionalidad frente a los cambios de iluminación. La combinación de LBP semánticas y la subdivisión de la escena en segmentos permitieron reducir la complejidad de la etapa de clasificación, aún a pesar de las deformaciones sufridas por los usuarios a medida que se alejan de la cámara. El uso de cámaras especiales no es necesario, cámaras web estándares cumplen plenamente las especificaciones técnicas mínimas para hacer el sistema confiable. La velocidad de fotogramas alcanzado fue por encima de 30 cuadros por segundo, suficiente para ser considerado un sistema en tiempo real. En el sistema de simulación, somos capaces de simular miles de agentes autónomos. El uso del modelo de fuerzas sociales en el lado de la simulación, produce un efecto plausible de interacción entre los dos mundos.

#### **4. Conclusión y trabajo futuro**

Hemos presentado un sistema interactivo que permite la interacción entre los usuarios peatonales y una multitud virtual, usando solo el hardware de consumo de grado. Este sistema es capaz de detectar y rastrear más de una docena de peatones en una escena, simulando miles de agentes virtuales y permitir la interacción entre los peatones y los agentes, manteniendo velocidades de escenas en tiempo real.

El sistema presentado en este trabajo puede ser útil para probar nuevos comportamientos de agentes virtuales, ya sean basadas en datos o basados en normas, ya que permite una comparación visual con peatones reales. También puede permitir simulaciones de planificación urbana y de seguridades más precisas, ya que una parte de la población en la multitud está constituida por personas reales.

En la industria del entretenimiento, este sistema proporciona una contribución al área de trabajo de aplicaciones que buscan la interacción entre el mundo real y un mundo virtual. Sobre la base de nuestro conocimiento, no hay juego en el mercado que utilice tanto la simulación de multitudes y la detección de multitudes para crear mundos y experiencias más inmersivas.

Como trabajo futuro se propone el empleo de dispositivos *wearables* como el *Google-Glass* para llevar el acoplamiento a la interacción permitiendo que las personas o usuarios reales puedan reaccionar a los personajes virtuales.

#### **Referencias**

1. Ahonen, T.: Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence* 28(12), 2037–41 (Dec 2006)
2. Banerjee, P., Sengupta, S.: Human motion detection and tracking for video surveillance. *Proceedings of the national Conference of tracking and video surveillance activity analysis* pp. 88–92 (2008)
3. van den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. *Robotics Research* 70, 3–19 (2011)

4. van den Berg, J., Manocha, D.: Reciprocal Velocity Obstacles for real-time multi-agent navigation. In: 2008 IEEE International Conference on Robotics and Automation. pp. 1928–1935. IEEE (May 2008)
5. Bhuvaneshwari, K., Rauf, H.A.: Edgelet based human detection and tracking by combined segmentation and soft decision. *Control, Automation, Communication and Energy Conservation* (June), 4–9 (2009)
6. 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, 886–893 (2005)
7. De Gyves, O., Toledo, L., Rudomín, I.: Comportamientos en simulación de multitudes : revisión del estado del arte. *Research in Computer Science*. 62(Special Issue: Avances en Inteligencia Artificial.), 319–334 (2013)
8. Groenewegen, S.: Improving crowd behaviour for games and virtual worlds. In: *Proceedings of the Fifth International Conference on the Foundations of Digital Games - FDG '10*. pp. 256–258. ACM Press, New York, New York, USA (2010)
9. Helbing, D., Farkas, I.J., Molnár, P., Vicsek, T.: Simulation of Pedestrian Crowds in Normal and Evacuation Situations. *Pedestrian and evacuation dynamics* 21 (2002)
10. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. *Physical Review E* 51(5), 4282–4286 (May 1995)
11. Huang, T.: Discriminative local binary patterns for human detection in personal album. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (Jun 2008)
12. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground-background segmentation using codebook model. *Real-Time Imaging* 11(3), 172–185 (Jun 2005)
13. Lengvenis, P., Simutis, R., Vaitkus, V., Maskeliunas, R.: Application Of Computer Vision Systems For Passenger Counting In Public Transport. *Electronics and Electrical Engineering* 19(3), 69–72 (Mar 2013)
14. Li, M., Zhang, Z., Huang, K., Tan, T.: Rapid and robust human detection and tracking based on omega-shape features. In: 2009 16th IEEE International Conference on Image Processing (ICIP). pp. 2545–2548. IEEE (Nov 2009)
15. Millan, E., Hernandez, B., Rudomin, I.: Large Crowds of Autonomous Animated Characters Using Fragment Shaders and Level of Detail. In: Wolfgang Engel (ed.) *ShaderX5: Advanced Rendering Techniques*, chap. Beyond Pix, pp. 501—510. Charles River Media (2006)
16. Moussaïd, M., Perozo, N., Garnier, S., Helbing, D., Theraulaz, G.: The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLoS one* 5(4), e10047 (Jan 2010)
17. Mukherjee, S., Das, K.: Omega Model for Human Detection and Counting for application in Smart Surveillance System. *arXiv preprint arXiv:1303.0633* 4(2), 167–172 (2013)
18. Ojala, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence* 24(7), 971–987 (2002)
19. Ondřej, J., Pettré, J., Olivier, A.H., Donikian, S.: A synthetic-vision based steering approach for crowd simulation. *ACM Transactions on Graphics* 29(4), 1 (Jul 2010)
20. Ozturk, O., Yamasaki, T., Aizawa, K.: Tracking of humans and estimation of body/head orientation from top-view single camera for visual focus of attention analysis. *Computer Vision Workshops (ICCV Workshops)* pp. 1020–1027 (Sep 2009)

21. Pelechano, N., Stocker, C.: Being a part of the crowd: towards validating VR crowds using presence. Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems (Aamas), 12–16 (2008)
22. Rivalcoba, I.J., Rudomín, I.: Segmentación de peatones a partir de vistas aéreas. In: Research in Computing Science. vol. 62, pp. 129–230 (2013)
23. Szymanczyk, O., Dickinson, P., Duckett, T.: From Individual Characters to Large Crowds : Augmenting the Believability of Open-World Games through Exploring Social Emotion in Pedestrian Groups. In: Think Design Play: DiGRA Conference (2011)
24. Treuille, A., Cooper, S., Popović, Z.: Continuum crowds. ACM Transactions on Graphics 25(3), 1160 (Jul 2006)
25. Tuzel, O., Porikli, F., Meer, P.: Human Detection via Classification on Riemannian Manifolds. 2007 IEEE Conference on Computer Vision and Pattern Recognition pp. 1–8 (Jun 2007)
26. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. International Conference on Computer Vision 63(2), 153–161 (2003)
27. Wang, Y., Dubey, R., Magnenat-Thalmann, N., Thalmann, D.: An immersive multi-agent system for interactive applications. The Visual Computer 29(5), 323–332 (Jun 2012)