



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MÉXICO



FACULTAD DE INGENIERÍA
MAESTRÍA EN CIENCIAS DE LA INGENIERÍA

GENERACIÓN DE MULTITUDES
HETEROGÉNEAS CON
COMPORTAMIENTOS INTELIGENTES EN
AMBIENTES VIRTUALES

TESIS

PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS DE LA INGENIERÍA

PRESENTA:
FERNANDO REBOLLAR CASTELAN

TUTOR ACADÉMICO:
DR. MARCO ANTONIO RAMOS CORCHADO

TOLUCA, ESTADO DE MÉXICO

2017

*Gracias a mis padres, familia, amigos, asesores y profesores por apoyarme, a la
Facultad de Ingeniería y a la UAEM, por la formación que me han dado, a
CONACYT por el respaldo económico.
Es gracias a ustedes que es posible el presente trabajo.*

Resumen

La simulación de multitudes consiste en representar a grupos de personajes autónomos llamados agentes virtuales, con reglas y entornos que simulan a aquellos encontrados en la vida real. Entre los objetivos de las simulaciones virtuales también se encuentra el generar algoritmos capaces de producir visualizaciones de manera fluida, esto es no tener retrasos en la visualización de la simulación. Las simulaciones de multitudes son herramientas utilizadas por organizaciones, por ejemplo una constructora, para realizar pruebas virtuales a determinadas estructuras como lo puede ser un estadio de fútbol, las multitudes utilizadas en este tipo de simulaciones carecen de comportamientos diferentes, es decir todos los individuos de la multitud se comportan de la misma manera aunque visualmente sean diferentes. La tesis presenta una propuesta para generar multitudes virtuales heterogéneas, de tal forma que los individuos que conforman la multitud exhiban distintos comportamientos, y no como se hace hasta el momento donde un mismo comportamiento que se aplica a todos los agentes. Se propone también una técnica de agrupamiento en distintas regiones del área a poblar utilizando diagramas de Voronoi, posibilitando la caracterización de zonas en donde existe una concentración selectiva dependiendo de los distintos roles que asumen los individuos de la multitud. Se introducen los LCP (Lugares de concentración de la población) que son los lugares donde distintos grupos de la población se agrupan, lo que genera un comportamiento de agrupación en las multitudes. Los resultados observados permiten distinguir distintos comportamientos debido a que se generan diferentes velocidades de desplazamiento, tamaños y pesos. La utilización de los LCP emula un comportamiento de agrupación dependiendo del rol que cada agente virtual tenga, lo cual permite que se agrupe en distintas zonas del ambiente virtual.

Abstract

Crowd simulation consists in representing groups of autonomous characters called virtual agents with rules and environments that simulate those found in real life. Among the objectives of virtual simulations, there is also the algorithms generator, which is able to produce visualizations in a stream way, this means not having delays in the simulation display. Crowd simulations are tools used by organizations, for example a builder, to perform virtual testing on endowed structures such as a football stadiums. Crowds that are used in those simulations lack different behaviors, for example, all individuals in the crowd behave in the same way, although they are visually different. This thesis presents a proposal to generate heterogeneous virtual crowds in a way where individuals that simulate the multitude exhibit different behaviors, as opposed to how it is done nowadays with a single behavior applying to all agents. It is also proposed a grouping technique in different places of the area of a village with Voronoi diagrams, making possible the characterization of areas where there is a selective concentration, depending on the different roles that individuals assume in the crowd. The LPC (Local People Concentration) is inserted in places where the population are grouped, which generates a grouping behavior in the multitudes. The observed results allow distinguish different behaviors due to different speeds of movement, size and weights. The use of LPC emulates a group behavior depending on the role that each virtual agent has, allowing it to be grouped in different areas of the virtual environment.

Tabla de Contenido

1. Introducción	1
1.1. Descripción del problema	1
1.2. Justificación	2
1.3. Objetivo general	3
1.4. Objetivos específicos	4
1.5. Estructura del documento	4
2. Marco teórico	5
2.1. Juegos Serios	5
2.2. Agentes virtuales	6
2.3. Criaturas Virtuales	7
2.4. Diagrama de Voronoi	8
2.5. Estado del Arte	10
3. Técnicas propuestas	19
3.1. Generación de comportamientos	19
3.1.1. El comportamiento humano	19
3.1.2. Velocidades de desplazamiento de los humanos	20
3.2. Roles en agentes y su distribución usando diagramas de Voronoi .	23
3.2.1. Lugares de concentración de población (LCP)	23
3.2.2. Vecino más cercano usando el diagrama de Voronoi	24
4. Resultados	29
5. Conclusiones y trabajo futuro	39
5.1. Conclusiones	39
5.2. Trabajo futuro	40
6. Publicaciones	41
Referencias	55

Índice de tablas

3.1. Fórmulas de velocidades de desplazamiento propuestas por Samson en metros sobre segundos [1]	21
3.2. Características y rangos utilizados en los agentes virtuales	22
3.3. Roles y rangos de edades utilizados en los agentes virtuales	27
3.4. Rangos de edades de la OMS	27
3.5. Roles y rangos de altura y peso redefinidos	28
4.1. Prueba de embotellamiento con 200 agentes virtuales	29
4.2. Prueba de búsqueda de ruta para llegar al objetivo	30
4.3. Roles y número de agentes virtuales por cada tipo de rol	34

Índice de figuras

1.1. Multitud de personas en avenidas públicas.[2]	3
2.1. Los sentidos y efectores permiten que el agente tenga interacción con el ambiente virtual.	7
2.2. Una criatura virtual consiste de una representación física que es la que se observa inmersa en el ambiente 3D.	8
2.3. Plano con un punto.	9
2.4. Diagrama de Voronoi con dos puntos en el plano.	9
2.5. Diagrama de Voronoi con tres puntos en el plano.	10
2.6. La parte de arriba muestra la simulación de los <i>boids</i> , referencia original de [3]. La imagen de abajo muestra los comportamientos de dirección <i>seguir</i> y <i>huir</i> , referencia original de [4]	12
2.7. Simulación VO de Fiorini [5]	14
2.8. Simulación de Cheney usando autómatas celulares [6]	14
2.9. Simulación fluida visualmente de D. Thalmann [7].	15
2.10. A la izquierda una simulación utilizando el algoritmo VO. A la derecha utilizando el algoritmo RVO [8].	16
2.11. Los agentes se mueven a la posición opuesta de donde comienzan en el círculo [9].	16
3.1. Plano con puntos (LCP) antes de calcular el diagrama de Voronoi	24
3.2. Plano con puntos (LCP), más un punto (agente virtual)	25
3.3. Diagrama de Voronoi del plano	25
3.4. Diagrama de Voronoi del plano con los LCP y un agente	26
3.5. Ambiente virtual dividido en distintas regiones de Voronoi [10]	27
4.1. Desplazamiento de agentes virtuales en un espacio que se reduce	30
4.2. Agentes virtuales con obstáculos	31
4.3. Multitudes intercambiando posiciones [10]	31
4.4. Patrones de multitudes obtenidas por los agentes [10]	32
4.5. Agentes virtuales heterogéneos	33

ÍNDICE DE FIGURAS

4.6. Agentes virtuales desplazándose por la ciudad buscando su LCP de interés [10]	33
4.7. Criaturas virtuales para representar a los agentes virtuales	35
4.8. Agentes virtuales caminando por la ciudad	35
4.9. Agentes virtuales sin sombras caminando por la ciudad	36
4.10. Agentes virtuales cerca de un LCP de su interés	37

Introducción

En este capítulo se presenta una descripción a la problemática, la justificación y los objetivos de la investigación, así como la estructura del documento.

1.1. Descripción del problema

El estudio de la concentración de individuos en lugares públicos como plazas, centros comerciales, parques, jardines, etc., es un campo de estudio abierto en distintas ciencias, como lo son las ciencias sociales y las ciencias formales, generando la necesidad de disponer de sistemas que permitan simular eventualidades en situaciones no controladas, como es en el caso de un terremoto. Disciplinas computacionales como la inteligencia artificial investigan cómo replicar el comportamiento humano de forma virtual, para obtener simulaciones que sean aproximadas a la realidad, apoyándose de las áreas sociales, psicológicas, neurocientíficas, entre otras.

Uno de los principales propósitos de realizar simulaciones de multitudes utilizando agentes virtuales es representar humanos virtuales autónomos [11] y que estas representaciones puedan exhibir comportamientos que se asemejen al de las personas. Otro objetivo es realizar simulaciones que se ejecuten de manera fluida, es decir que visualmente la simulación debe garantizar una frecuencia de al menos 30Hz [12]. A pesar de que el problema de la simulación de multitudes lo han estudiado especialistas en gráficas computacionales, diversas áreas se han sumado para crear simulaciones cada vez más complejas como físicos y especialistas en Inteligencia Artificial (IA), entre otros, han creado modelos para simular multitudes con cada vez más agentes virtuales [13].

No existe un número determinado de personajes que constituyen una multitud, sin embargo, los investigadores a menudo tratan de aumentar el número

tanto como sea posible, desde unos pocos cientos hasta millones, pero siempre esforzándose por mantener la meta de visualizaciones fluidas en la simulación [14].

Las aproximaciones actuales para la generación de multitudes son centradas en las reglas del ambiente físico y el comportamiento está basado en la interacción con el ambiente y las reglas físicas del mismo [15, 16].

Thalman en [17] define los problemas comunes en las simulaciones de multitudes, y dice que los principales son: “Diversidad visual, animación de personajes, cantidad de personajes, inteligencia artificial y la planificación de rutas”.

Las investigaciones existentes [6, 18, 19, 20] tienen carencias debido a que tienen la imposibilidad de poder observar comportamientos de tipo individual en la multitud virtual, aunado a esto, se observa que las multitudes carecen de diversidad en las poblaciones debido a la carencia de comportamientos heterogéneos [21].

Ruiz en [22] destaca la importancia de que los algoritmos que se generen para dotar de comportamientos a las multitudes virtuales deben contar solamente con los elementos necesarios para que los algoritmos no impidan una simulación fluida visualmente, además de que los comportamientos humanos son individuales y grupales, por ello es necesario considerar ambos para dotar a una multitud virtual.

Antes de reproducir comportamientos individuales y grupales existen dos enfoques que deben abordarse: la planificación y la toma de decisiones. La planificación dotará a los agentes para poder observar el medio ambiente, para posteriormente decidir qué acción tomar sobre las condiciones del mismo, por ejemplo la evasión de colisiones. Los métodos principales para la evasión de colisiones se basan en fuerzas sociales [23] mientras que la velocidad es manejada de manera recíproca para todos los agentes. Un algoritmo simple para evitar colisiones compara la posición de cada agente con los demás, sin embargo, la complejidad del algoritmo es $O(N^2)$, cada agente virtual debe comparar su posición con los demás agentes virtuales y conforme el número de agentes (denotado por N) crece, dicha complejidad representa un problema cuando se aumenta el número de agentes en las multitudes e imposibilita la obtención de una simulación fluida visualmente.

1.2. Justificación

El comportamiento colectivo en los humanos se ha estudiado desde principios del siglo XIX con la finalidad de observar las acciones que las personas realizan cuando se presentan diferentes circunstancias como celebraciones, manifestaciones, simulacros o incluso cuando ocurren fenómenos naturales como incendios o terremotos [24]. Por otro lado, investigaciones en ciencias computacionales realizan

esfuerzos para simular de manera virtual multitudes con el objetivo de reproducir comportamientos similares a los seres humanos con la finalidad de estudiar eventualidades en las multitudes [12].

Las simulaciones de multitudes son realizadas a partir de diferentes enfoques dependiendo el ambiente virtual que se desea poblar, por ejemplo: la industria del entretenimiento (videojuegos [25, 26, 27], películas [12], realidad virtual [28], etc.). En los últimos años este tipo de simulaciones ha tomado importancia en el uso de los llamados *juegos serios* [29]. Éstos requieren ser poblados con entidades virtuales (agentes) [30] que permitan al usuario contar con una retroalimentación o ser asistido en tareas inmersas en el ambiente virtual, como por ejemplo sistemas de entrenamiento, sistemas basados en simulacros (incendios, terremotos, manifestaciones, guerrillas, etc.) [31].



Figura 1.1: Multitud de personas en avenidas públicas.[2]

Este tipo de simulaciones también es usada para la seguridad pública, en lugares y situaciones en donde densos grupos de personas se ven involucradas, como se observa en la Figura 1.1. Los gobiernos pueden hacer uso de las simulaciones de multitudes con el fin de obtener información que ayude a prevenir situaciones de desastre en actos públicos, aeropuertos y pasos subterráneos [32, 33, 34, 35, 36, 37, 38].

1.3. Objetivo general

Poblar un entorno virtual con multitudes que exhiban comportamientos heterogéneos y formen patrones de agrupación en determinados lugares dependiendo del rol que asuma cada agente virtual, que permitan realizar simulaciones fluidas visualmente.

1.4. Objetivos específicos

1. Proponer una técnica para que las multitudes tengan comportamientos heterogéneos, que permita dotar a porciones de la multitud con roles de niños, jóvenes, adultos y ancianos.
2. Proponer una técnica para la distribución de la población en el ambiente virtual de acuerdo a las características de distintas zonas.
3. Embeber las multitudes en un motor físico computacional para realizar simulaciones y para observar los comportamientos de las multitudes.

1.5. Estructura del documento

El documento cuenta con 6 capítulos, este capítulo es una breve introducción al trabajo realizado. La organización de los siguientes capítulos se describe a continuación.

El Capítulo 2 contiene el marco teórico con conceptos relacionados con la investigación así como una revisión del estado del arte.

El Capítulo 3 describe la forma en que se generan comportamientos heterogéneos en los agentes así como una clasificación de los mismos y de los distintos roles que representan, también describe la técnica propuesta para generar patrones de agrupación de los agentes en distintas zonas del ambiente virtual.

El Capítulo 4 muestra los resultados obtenidos al realizar simulaciones en un motor gráfico.

El Capítulo 5 contiene las conclusiones de la presente investigación así como el trabajo a futuro de la investigación.

El Capítulo 6 es el artículo publicado derivado de la investigación.

En este capítulo se describen conceptos necesarios que fundamentan la investigación, así como también algunos trabajos relacionados.

2.1. Juegos Serios

El juego es una actividad fundamental para el desarrollo humano. Comúnmente se juega para divertirse, para entretenerse; sin embargo, hay quien afirma que se juega principalmente para aprender, aunque ésta sea una intención inconsciente [39]. Los juegos serios según Michael y Chen (2006) [40] son aquellos juegos que se usan para la educación, entrenamiento e información. El término juego serio se ha utilizado desde la década de los 60s cuando realizaban juegos basados en guerras pasadas como la primera guerra mundial y se realizaban simulaciones de ataques para que se debatieran estrategias de guerra. Sin embargo, en el siglo XXI se da ese nombre a los videojuegos y/o herramientas tecnológicas como lo pueden ser simuladores de vuelo de diversos tipos de aviones, donde el entretenimiento queda en segundo plano y se utilizan para desarrollar o mejorar distintas habilidades y destrezas, es decir que se busca obtener uno o varios beneficios del juego. Esta área de desarrollo y creación de videojuegos ha surgido como una manera inteligente de combinar los beneficios de los videojuegos. Su creciente utilidad ha llamado la atención en la población y las distintas necesidades han llevado su uso a nivel político-institucional como empresarial y comercial [41].

La idea central de los juegos serios es infundir en una simulación, algún tipo de aprendizaje, así como entrenamiento con la interacción en el videojuego, para mejorar habilidades [42]. Los juegos serios pueden potencialmente motivar la práctica regular necesaria para el desarrollo de habilidades y competencias de forma sencilla y sistemática, sin descuidar la experiencia de los jugadores como:

la diversión, inmersión, presencia, reto, curiosidad y otras emociones [43]. Una característica de los juegos serios es que esta relacionada con algún aspecto de la realidad. Por ejemplo, un área que se beneficia substancialmente es el sector salud con juegos serios de entrenamientos y monitoreo de los pacientes entre otros. El creciente aumento de la población y la falta de un adecuado número de personal capacitado puede generar un déficit de adecuada atención a los pacientes que al no tener la atención correcta y sufren efectos negativos en su desarrollo físico, social y psicológico, por ejemplo en fisioterapia y rehabilitación, donde es necesario que los pacientes realicen de manera consistente y regular una serie de ejercicios físicos, que requieren la supervisión de una persona especializada sobre un largo periodo de tiempo [44], utilizando un juego serio que permita programar la serie de ejercicios personalizados y que el paciente pueda reportar sus avances por medio del juego, el especialista puede atender a un mayor número de pacientes y programar nuevos ejercicios a los mismos incluso remotamente puesto que algunos pacientes pueden completar los ejercicios en la comodidad de su hogar, donde desde el punto de vista psicológico el paciente se siente mucho mejor [45]. Otra área pionera en los juegos serios es la militar, que ha desarrollado una variedad de simuladores para entrenar a los soldados, para mejorar sus habilidades en el campo de batalla y para utilizar la variedad de armamento con el que se cuenta en la actualidad (aviones, helicópteros, barcos, submarinos, misiles, etc.). Sin embargo, en otras poblaciones y debido a cuestiones religiosas y culturales en algunos casos, la implementación de estos juegos serios aún es un impedimento, por ejemplo los *Amish* [46] que rechazan cualquier tipo de tecnología.

2.2. Agentes virtuales

Existe en la literatura una diversidad de definiciones en cuanto se refiere a un agente virtual, cada definición se adapta a los diferentes propósitos para los cuales se utilizan los agentes virtuales, es por eso que para la presente investigación se utilizará la siguiente definición.

“Los *agentes virtuales* son sistemas autónomos que se utilizan en una computadora situados en un ambiente virtual, capaces de actuar de manera autónoma para cumplir con los objetivos para los cuales fueron diseñados” [47]. La interacción de un agente con su ambiente se da gracias a sus sentidos y efectores (ver Figura 2.1). Mediante los sentidos, el agente es capaz de reconocer objetos dentro del mismo ambiente, así como también de mantener un estado de alerta sobre los cambios en el ambiente realizados por él mismo, o por una entidad externa al agente con lo cual es capaz de tomar decisiones y retroalimentar su conducta para cumplir con sus objetivos. Los objetivos de los agentes son programados por

medio de la base de conocimientos de los agentes que les permite reaccionar a los objetos que se encuentren en el ambiente.



Figura 2.1: Los sentidos y efectores permiten que el agente tenga interacción con el ambiente virtual.

En un sistema multiagentes el ambiente se define como el conjunto $E = \{e_0, e_1, \dots, e_u\}$ donde E es el ambiente conformado por todos los posibles estados. $Ac = \{\alpha_0, \alpha_1, \dots, \alpha_u\}$ representa todas las acciones permitidas dentro de E . Los agentes construyen su base de conocimientos a partir de $r : e_0 \xrightarrow{\alpha_0} e_1 \xrightarrow{\alpha_1} e_2 \xrightarrow{\alpha_2} e_3 \xrightarrow{\alpha_3} \dots \xrightarrow{\alpha_{u-1}} e_u$ que representan las acciones realizadas de un estado a otro dentro de E , por ejemplo el estado e_1 con una acción α_1 permite al agente pasar a un estado e_2 .

Un agente virtual puede obtener una representación visual por medio de una criatura virtual la cual se describe a continuación.

2.3. Criaturas Virtuales

La criatura virtual es el ente informático que está inmerso en un ambiente virtual \mathbb{R}^3 y consiste de una representación física y de inteligencia. La parte física es un modelo en tres dimensiones que generalmente representa a un ser vivo (ver Figura 2.2). La inteligencia de la criatura es producida por un agente virtual, dicha inteligencia le permite desarrollarse en el ambiente para alcanzar las metas que se le encomiendan [11].



Figura 2.2: Una criatura virtual consiste de una representación física que es la que se observa inmersa en el ambiente 3D.

2.4. Diagrama de Voronoi

El diagrama de Voronoi es un método matemático en el cual se crea una región por cada punto que exista en el plano, se genera a partir de las mediatrices de los segmentos que forman los puntos que se encuentren en el plano.

Sea $P = \{p_1, p_2, \dots, p_n\}$ un conjunto de puntos en el plano, haciendo uso del diagrama de Voronoi es posible asignar a cada punto una región en el plano correspondiente a los puntos más cercanos. Para cada $p_i \in P$, forman la región de Voronoi $V(p_i)$ [48]. Dado un conjunto de puntos P y un punto de consulta q , es posible determinar el punto más cercano a q en P , dado que la ubicación de q está dentro de una región de Voronoi en un punto p_i la cual indica que dicho punto p_i es el más cercano al punto q . Los diagramas de Voronoi codifican información de proximidad entre elementos, los que podemos tratar como celdas $V(p_i)$ es decir, la celda de Voronoi p_i se define como:

$$V(p_i) = \{q / \|p_i q\| < \|p_j q\|, \forall j \neq i\} \quad (2.1)$$

Donde $\|p_k q\|$ es la distancia euclídea entre p_k y q con $k = i, j$.

Como primera aproximación podemos decir que dado un conjunto P de sitios (puntos) en el plano, su diagrama de Voronoi es la partición de ese plano en regiones (una región para cada sitio), tal que la región del sitio p contiene todos

los puntos del plano que están más cerca de p que de cualquier otro sitio en P .

El diagrama de Voronoi es una construcción geométrica que permite asignar a cada punto una región, de forma que todo lo que se encuentra en esa región, tiene como punto más cercano al punto de la región de Voronoi que de cualquier otro, y se traza colocando tantos puntos como se quiera en un espacio \mathbb{R}^2 aleatorio y se tiene que trazar una línea recta entre los dos puntos, al principio se tiene un punto (ver Figura 2.3):

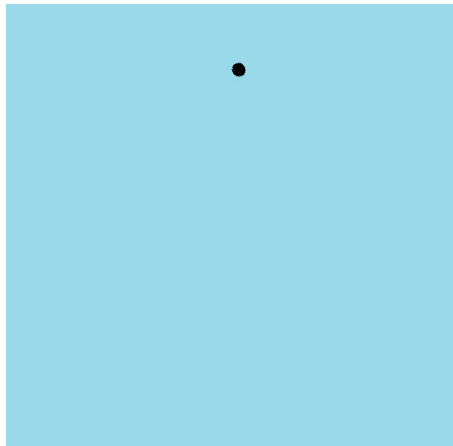


Figura 2.3: Plano con un punto.

Al colocar otro punto se traza una recta entre los dos puntos que se calcula haciendo la mediatriz del segmento de los mismos y por el punto medio se traza una línea perpendicular. (ver Figura 2.4).

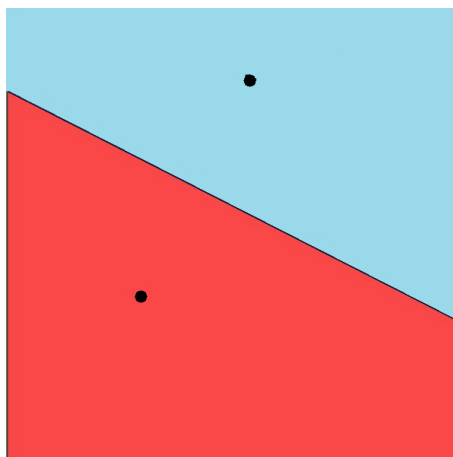


Figura 2.4: Diagrama de Voronoi con dos puntos en el plano.

Al agregar otro punto en el plano el proceso es igual, se traza una mediatriz con respecto a los segmentos que unen los otros dos puntos para delimitar su región (ver Figura 2.5).

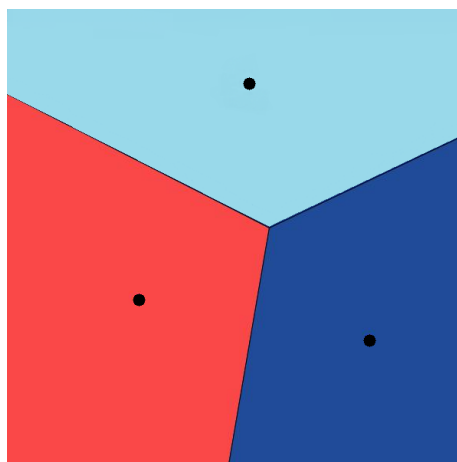


Figura 2.5: Diagrama de Voronoi con tres puntos en el plano.

Al agregar más puntos en el plano se siguen realizando los mismos cálculos hasta que cada punto tiene una región delimitada correspondiente a la distancia en que se encuentren sus vecinos.

Existe una serie de algoritmos para construir el diagrama de Voronoi de un conjunto de n sitios en el plano. En la literatura se puede encontrar un algoritmo trivial con la complejidad en $O(n^2 \log n)$ y que opera construyendo p_i mediante la intersección de $n - 1$ semiplanos $h(p_i, p_j)$, para $2 \leq j \leq n$, $j \neq 1$.

Posteriormente se desarrollaron algoritmos para construir el diagrama de Voronoi con menor complejidad, siendo el más utilizado el algoritmo de Steven Fortune en 1987 [49] quien inventó un algoritmo de barrido de plano que se ejecuta en tiempo $O(n \log n)$.

2.5. Estado del Arte

Las multitudes son parte de nuestra experiencia cotidiana, han existido desde los inicios de la humanidad; sin embargo, en los mundos virtuales son todavía poco comunes. En 1987 Craig Reynolds fue quien realizó uno de los primeros trabajos sobre la creación de multitudes, basándose en el comportamiento de las aves y la manera de cómo se agrupan durante el vuelo [3]. Un ave realiza movimientos simples pero cuando se agrupan muchas aves los movimientos se vuelven visualmente complejos, Reynolds hace uso de tres reglas simples: separación, alineación

y cohesión, por orden de importancia. Según los resultados de Reynolds, un pájaro durante el vuelo en grupo puede distinguir su misma presencia, la de algunos vecinos y es consciente de pertenecer a una bandada, estos tres elementos hacen posible que un ave se identifique como parte de un grupo sin importar el número de integrantes del grupo.

Reynolds introdujo el término *boids*, para referirse a los agentes que siguen las reglas de movimiento que él propuso. En 1999 Reynolds amplía sus reglas de movimientos, en su investigación [4] explica que cuando se agregan nuevas reglas de movimientos los agentes son capaces de realizar cambios de direcciones, permitiendo observar comportamientos como los observados en las parvadas de aves. Reynolds nombra estas reglas como comportamientos de dirección (*steering behaviors* en inglés), y menciona que combinar distintos comportamientos de dirección genera patrones de comportamientos más complejos. En la Figura 2.6 se observa en la parte de arriba la simulación de Reynolds y en la parte de abajo se muestran algunos comportamientos de dirección.

Se considera un comportamiento reactivo a todas aquellas acciones que son desencadenadas mediante un evento, es decir mientras no se registre ningún estímulo del ambiente o incluso de algún otro agente involucrado en la simulación. Los agentes que pertenezcan a la categoría reactiva no tienen ninguna razón para reaccionar [12]. Para esto Erik Millan et al. utilizan XML y generan archivos con máquinas de estado finitas, dichos archivos los guardan en formato de imagen y los agentes consultan estas imágenes para generar comportamientos [14]. Pelechano y Badler [20] han combinado reglas de percepción y comportamientos, aunque son comportamientos simples para dirigir agentes en entornos virtuales, si estos son combinados adecuadamente se pueden obtener nuevos comportamientos que parecen complejos.

Las simulaciones donde el ambiente virtual es el encargado de seleccionar la mejor acción de los agentes están basadas en reglas que no requiere razonamiento complejo, la investigación presentada por Wenhua en [50] utiliza un ambiente virtual que se encarga de seleccionar las acciones de los agentes, sin embargo esto se aleja de lo que sucede en la realidad puesto que el ambiente no controla el movimiento de las personas en una multitud solo los limita, ya que éstos se desplazan libremente. Kapadia en [19] utiliza comportamientos reactivos, y movimientos de dirección en plataformas que dirigen a los agentes pudiendo realizar predicciones de espacio-tiempo en las rutas de los agentes para evitar que choquen.

Las personas evitan chocar con objetos y con otras personas estando al tanto de su entorno, donde el entorno es dinámico y es necesario que la persona este actualizando su información sobre la ubicación de los objetos que se encuentren en el ambiente, esto le permite adaptar su trayectoria continuamente y con ello atravesar un camino con obstáculos dinámicos. Lo mismo se busca replicar con los agentes virtuales, se busca que los agentes tengan conocimiento de los objetos

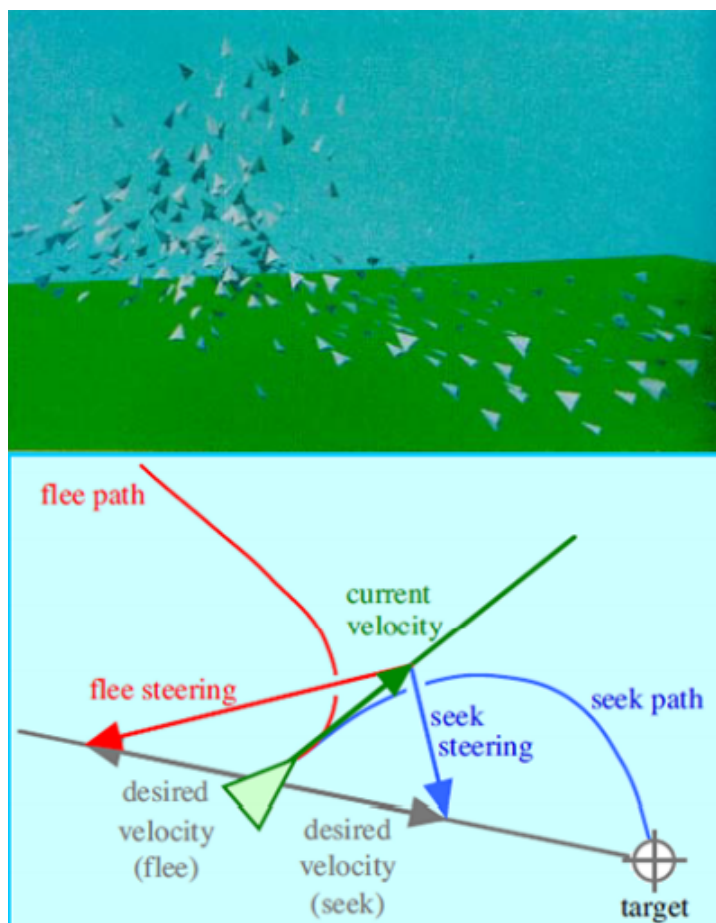


Figura 2.6: La parte de arriba muestra la simulación de los *birds*, referencia original de [3]. La imagen de abajo muestra los comportamientos de dirección *seguir* y *huir*, referencia original de [4]

que lo rodean para que el mismo pueda ser capaz de modificar su ruta y eso le permita llegar a su objetivo [12].

Una de las formas de resolver colisiones en simulaciones de agentes virtuales ha sido mediante la aplicación de algoritmos de búsquedas de proximidad, donde un algoritmo básico tiene una complejidad de $O(N^2)$ donde N es el número de agentes [22]. Esto significa que cada agente tiene que consultar la posición de todos los demás agentes, sin contar los otros objetos que pudiese haber en su entorno. Estructuras de datos especiales surgieron para tratar de resolver este problema, reducir la complejidad de las búsquedas de proximidad y permitir simulaciones de grandes multitudes que se miren visualmente fluidas.

En [51] se utilizan estructuras jerárquicas, como los *octrees*, donde el espacio se subdivide en varias regiones de un mismo tamaño, y que cada región contiene

a un número determinado de agentes que tienen conocimiento de los vecinos que pertenecen a la misma región. Algunos trabajos como [8] y [52] utilizan *árboles kd* para consultar a los vecinos más cercanos de manera más eficiente. Existen también implementaciones en paralelo de *árboles kd*, algunas enfocadas en la unidad de procesamiento central (CPU) [53] y otras enfocadas en el procesador gráfico (GPU) [54]. Bleiweiss en [55] presentó una implementación en paralelo de la biblioteca popular para evasión de colisiones, obteniendo un aumento de velocidad de 4.8X en comparación con la implementación original. Bleiweiss cambió el método de búsquedas de proximidad de *árboles kd* a un método basado en hash con el fin de mejorar el rendimiento en el GPU. Jund en [56] presentó una estructura de datos topológica de resolución múltiple para la búsqueda de vecinos, que codifica la información para cada agente, en una estructura celular de manera que cada agente consulta únicamente a sus vecinos adyacentes al agente, por lo que el número de cálculos es menor al método anterior. Otra forma de encontrar a los vecinos más cercanos consiste en almacenar a los agentes dentro de una matriz ordenada como lo hace Passos et al. en su investigación [57] donde utiliza la matriz ordenada y usa la técnica de vecindario extendido de Moore (*extended Moore neighborhood* en inglés) para encontrar a vecinos más próximos al agente.

Hoff et al. en [58], presentaron una técnica que guarda las coordenadas de las ubicaciones de los agentes en un plano de dos dimensiones, con las cuales calculan con ecuaciones geométricas las distancias con los de mas agentes, también hacen una marcación de los objetos de interés y marcando los píxeles que pertenecen a los objetos para identificarlos. Millan propuso una técnica que permitía a los agentes moverse por distintas zonas y los agentes al mismo tiempo que se movían pintaban las áreas por donde pasaban de tal manera que cada agente pintaba un área de influencia de su posición la cual es leída por otros agentes para moverse a través del ambiente virtual por las zonas menos transitadas evitando encontrarse con otros agentes [14].

Fiorini y Shiller en [5], introducen el término *Velocity Obstacles* (VO), una técnica geométrica para detectar objetos que van a chocar si sus rutas no son modificadas, asumiendo que las velocidades de los objetos no cambian. VO es el conjunto de todas las velocidades que conducirán eventualmente a dos agentes a una colisión, por lo que éstos deben elegir velocidades que no sean elementos del conjunto que es clasificado como velocidades que llevan a colisiones. El algoritmo genera caminos libres de colisión, pero con un gran número de oscilaciones, que hacen al movimientos bruscos, ver Figura 2.7.

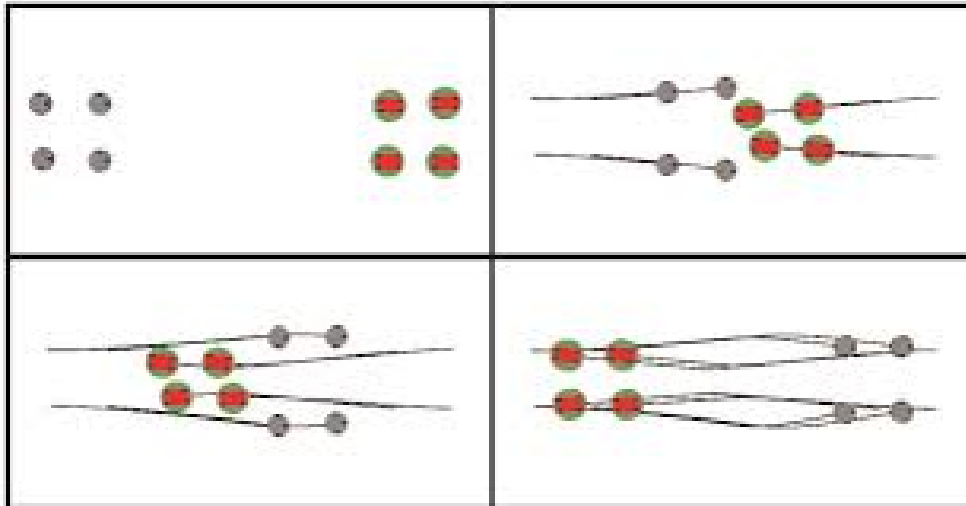


Figura 2.7: Simulación VO de Fiorini [5]

En 2004, Cheney [6] propone una técnica para la representación y el diseño de campos de velocidad, usando autómatas celulares. Los campos de velocidad creados usando esta técnica, son útiles para crear movimientos de flujo, que son seguidos por los peatones, con el fin de moverse a través de un entorno (ver Figura 2.8).



Figura 2.8: Simulación de Cheney usando autómatas celulares [6]

Treuille presenta el modelo *Continuum Crowds* en 2006, en el cual unifica la planeación de rutas y la evasión de colisiones utilizando campos potenciales y de

velocidad para guiar a los agentes virtuales [59]. En este modelo se logran conseguir simulaciones visualmente fluidas, movimiento uniforme y comportamientos emergentes. En 2009 Fischer utiliza campos potenciales para permitir que los agentes negocien el espacio, evitar colisiones y alcanzar sus metas [60], el algoritmo es una versión paralela de una técnica anterior de los mismos autores [61]. La simulación se realizó en el GPU utilizando la plataforma *Compute Unified Device Architecture* (CUDA) de Nvidia.

D. Thalmann en 2006 [21], determinó las principales razones que dificultaban el uso de las multitudes virtuales. Una de las principales razones son las altas exigencias en los recursos computacionales, hablando en términos de procesamiento tanto en calculo y visualización, otra razón es el costo de producción de contenidos para ser utilizados en las simulaciones. Gracias a que la tecnología cambia rápidamente, el rendimiento del hardware se ha multiplicado en los últimos años, lo que hace posible mostrar escenas virtuales en 3D con miles de entidades individuales animadas que anteriormente no era posible. En 2006, Pettre y Thalmann [7] construyeron un simulador con miles de agentes donde la multitud se movía de un lugar a otro con una visualización fluida (ver Figura 2.9).

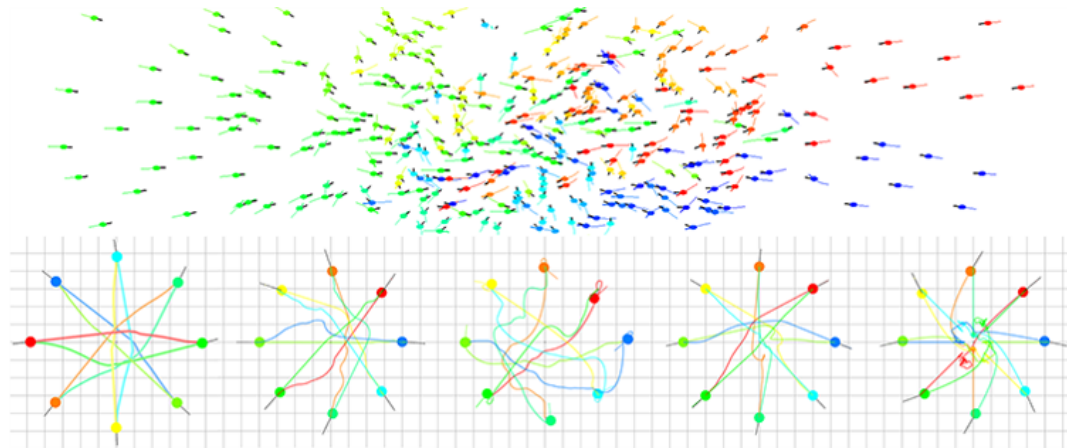


Figura 2.9: Simulación fluida visualmente de D. Thalmann [7].

Hubert hace uso de autómatas celulares en la simulación de grandes eventos [62]. En este modelo, las celdas también son una representación discreta del espacio geográfico en el que los agentes son capaces de moverse, aunque representan un área en lugar de ranuras disponibles, por lo que los agentes pueden moverse a través de ellas, no solo de una a otra.

En 2008 Van den Berg retoma la idea propuesta por Fiorini en Reciprocal Velocity Obstacles (RVO) [8]. Berg, hace dos mejoras al algoritmo de Fiorini; la primera es modificando las rutas que siguen los agentes incorporando el paso de mensajes a otros agentes (ver Figura 2.10) para así lograr los objetivos. La

2. MARCO TEÓRICO

segunda mejora agrega un método para indicar la distancia en la que un agente debe cambiar su trayectoria original para evitar colisionar con otros agentes. De esta manera se crean las prioridades de movimiento de los agentes para el logro de sus metas.

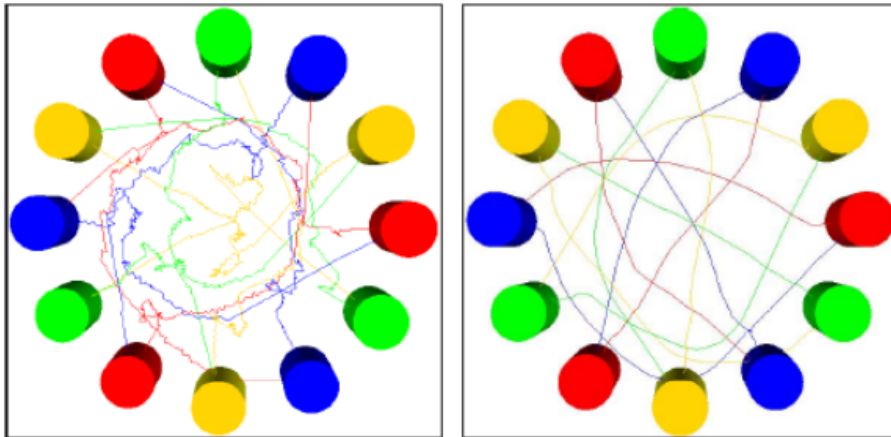


Figura 2.10: A la izquierda una simulación utilizando el algoritmo VO. A la derecha utilizando el algoritmo RVO [8].

El mismo grupo de investigación de Van den Berg amplía aún más el algoritmo de RVO, reduciendo el problema a un programa lineal de baja dimensión [9]. Mediante el uso de este algoritmo, los autores fueron capaces de simular miles de personajes, aún en CPU. En la Figura 2.11 se observa una prueba visualmente fluida con 1,000 agentes virtuales.

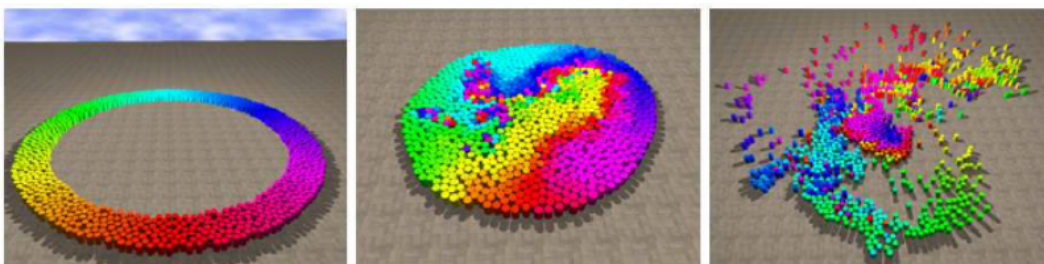


Figura 2.11: Los agentes se mueven a la posición opuesta de donde comienzan en el círculo [9].

En 2011 Zhang [63] presenta un modelo en el que las celdas de un autómata celular, representan posiciones discretas en el espacio, utilizadas por los peatones para moverse al cambiar de una celda a otra. Nuevamente Thalmann presenta un sistema de navegación interactiva para multitudes en 2013 [64]. Su sistema

permite a un usuario seleccionar un agente de entre la multitud y controlar su movimiento a través del entorno virtual, por medio de una interfaz natural que captura los movimientos del usuario.

En 2015, fue presentado un sistema capaz de simular grandes multitudes que también considera pequeños grupos de agentes relacionados por Rudomín [13]. Este sistema tiene una arquitectura paralela. De esta manera se han conseguido simulaciones visualmente fluidas con miles de agentes, dicha simulación utiliza la técnica de diagramas de Voronoi truncado del mismo grupo de trabajo de Rudomín [65].

Como se puede notar en el estado del arte existe la carencia de encontrar distintos comportamientos en las multitudes, si bien existen trabajos que permiten simulaciones visualmente fluidas de multitudes en el entorno de miles de agentes virtuales estos agentes tienen un mismo comportamiento, es por eso que en el siguiente capítulo se presenta una técnica para obtener comportamientos heterogéneos dentro de una multitud.

Técnicas propuestas

En este capítulo se describen dos técnicas propuestas en la investigación, la primera para generar diferentes comportamientos en la multitud y la segunda para hacer que la multitud se agrupe en distintos espacios dentro del ambiente virtual.

3.1. Generación de comportamientos

Un problema a abordar en la generación de multitudes es la carencia de diversidad de comportamientos en los agentes virtuales que la conforman, debido a que sólo se utiliza un modelo del agente que se replica de manera indiscriminada, cambiando sólo el color de la ropa o cabello. También se puede observar la falta de autonomía en los agentes ya que todos los agentes de la simulación tienen un mismo comportamiento lo cual reduce el realismo en la simulación.

La propuesta se basa en la generación de multitudes heterogéneas a través de diversos modelos que conforman la multitud sin descuidar los comportamientos asociados a cada agente. Por ejemplo, si en la escena se observa a un niño, este tiene una altura, peso y velocidad de desplazamiento menor en comparación con un adulto, y si se encontraran en ruta de colisión el agente de menor peso es el que modifica su ruta mientras que el agente de mayor peso mantiene la misma ruta.

3.1.1. El comportamiento humano

El comportamiento de cada persona en la vida real es único, el cual está determinado por un sin fin de factores que lo van determinando a lo largo del tiempo y de experiencias vividas por cada persona. También lo delimitan aspectos fisiológicos

3. TÉCNICAS PROPUESTAS

y capacidades que impiden poder realizar algunas actividades para determinados sectores de la población.

Una de muchas definiciones que existen del comportamiento humano dice que es el conjunto de actos exhibidos por el ser humano y determinados por la cultura, las actitudes, las emociones, los valores de la persona, los valores culturales, la ética, el ejercicio de la autoridad, la relación, la hipnosis, la persuasión, la coerción y/o la genética [66].

El comportamiento humano es considerado un tema complicado, el ser humano ha demostrado su interés de aprender sobre lo que lo rodea y aprovecharlo para su beneficio y comodidad, si bien el ser humano es curioso, también es creativo al inventar toda una serie de formas para comunicarse, desde el lenguaje por señas, el escrito, el oral o por medio de sonidos, incluso el comportamiento corporal, entre otras muchas más cosas que ayudaron a facilitar la vida, así como su supervivencia. Otro aspecto importante sobre la forma de ser de las personas es la manera en la que éstas aprenden, pero solo aprenden aspectos determinados lo que determinar qué aprender y qué no aprender, y esto a su vez es influenciado por los procesos de atención, lo cual posee mucha relevancia ya que desde la infancia se fomenta el imitar como una manera de aprender, así se aprende a hablar, caminar entre otras cosas, de esta manera muchos adoptan la personalidad de otra persona lo que ocasiona que existan comportamientos similares entre grupos de personas con características parecidas [67].

Lo ideal en las simulaciones de multitudes es que cada agente se comporte de manera diferente tal y como pasa en la vida real. Sin embargo, este proceso tomaría demasiados recursos del sistema para que cada agente virtual contara con un comportamiento individual impidiendo su simulación visualmente fluida. De ahí la necesidad de delimitar las diferentes variables a considerar para cada tipo de agente, por lo que se busca generar con el menor número de variables distintos comportamientos para cada tipo de agente virtual.

3.1.2. Velocidades de desplazamiento de los humanos

Existen distintas velocidades a las que se desplazan las personas utilizando sus piernas. Dependiendo de las distintas necesidades y de las distintas capacidades que tengan las personas es como se obtendrán las distintas velocidades en que las personas se pueden desplazar. En 2001 Samson [1] calculó la velocidad de desplazamiento promedio de diferentes personas, determinó la cadencia de los pasos y longitud de zancada de las personas, considerando parámetros como la edad, el peso y la altura de las personas. Para ello, analizó a 118 mujeres y 121 hombres en un rango de entre los 19 y los 90 años de edad, los cuales caminaban a su velocidad preferida como lo harían normalmente a través de una pasarela

de 12 metros. Samson después de analizar todos los resultados que recopiló en sus pruebas propone fórmulas para calcular la velocidad de desplazamiento de los individuos en actividades cotidianas. En nuestro caso haremos uso de algunas fórmulas propuestas por Samson para embeberlas en los agentes virtuales para reproducir estos desplazamientos con la característica de que se puede distinguir hombres y mujeres dentro de la simulación. La Tabla 3.1 muestra la relación de velocidad desplazamiento existente entre hombres y mujeres.

Tabla 3.1: Fórmulas de velocidades de desplazamiento propuestas por Samson en metros sobre segundos [1]

Hombres V en m/s
1.- $V = 1.460$
2.- $V = -0.002 (\text{edad}) + 1.582$
3.- $V = -0.002 (\text{edad}) + 0.442 (\text{altura}) + 0.750$
4.- $V = -0.001 (\text{edad}) + 0.486 (\text{altura}) - 0.001 (\text{peso}) + 0.720$
Mujeres V en m/s
5.- $V = 1.420$
6.- $V = -0.003 (\text{edad}) + 1.552$
7.- $V = -0.002 (\text{edad}) + 0.618 (\text{altura}) + 0.484$
8.- $V = -0.001 (\text{edad}) + 0.827 (\text{altura}) - 0.003 (\text{peso}) + 0.316$

En los resultados de la investigación se llega a la conclusión que los hombres en promedio se desplazan a una velocidad de 1.46 metros por segundo y que las mujeres se desplazan en promedio a 1.42 metros por segundo, pero si se considera la edad de los hombres la fórmula que se utiliza es la segunda que dice que en promedio los hombres se desplazan a una velocidad en metros por segundo que se obtiene de multiplicar la edad por -0.002 y sumarle 1.582, en el caso de las mujeres considerando su edad la velocidad de desplazamiento promedio se obtendría multiplicando la edad por -0.003 y sumarle 1.552, las cuales son las fórmulas 2 en el caso de los hombres y 6 en el caso de las mujeres.

También es posible calcular la velocidad de desplazamiento promedio considerando la altura además de la edad de las personas. En el caso de los hombres esta se obtiene con la fórmula 3 y en el caso de las mujeres con la fórmula 7, pero mejor aún y para obtener mejores resultados se utiliza en los agentes las fórmulas 4 (para el caso de los hombres) y 8 (para el caso de las mujeres), que permiten considerar la edad, altura y el peso de las personas para calcular sus velocidades promedio distinguiendo entre hombres y mujeres.

Por ejemplo, para conocer la velocidad de desplazamiento promedio de un individuo masculino con una edad de 49 años, altura de 1.8 metros y peso de 90 kilogramos, su velocidad se obtendrá utilizando la fórmula 4, sustituyendo

3. TÉCNICAS PROPUESTAS

obtenemos:

$$V = -0.001(49) + 0.486(1.8) - 0.001(90) + 0.720$$

$$V = -0.049 + 0.8748 - 0.09 + 0.720$$

$$V = 1.4558 \text{ m/s.}$$

Para calcular las velocidades de los agentes estos son dotados de una determinada edad, altura, peso y sexo, estos valores son aleatorios y tienen distintos rangos los cuales se muestran en la Tabla 3.2, en el caso del sexo solo se consideran dos valores posibles, hombre o mujer.

Tabla 3.2: Características y rangos utilizados en los agentes virtuales

Característica	Rango
Edad en años	6 - 99
Altura en metros	0.90 - 2.00
Peso en kilogramos	25 - 120

Se considera la edad en años, la altura en metros y el peso en kilogramos, estas mismas unidades fueron consideradas por Samson en su investigación.

La posibilidad de dotar a los agentes de valores aleatorios en el desplazamiento, así como condicionantes de masa y altura permite obtener en la simulación de multitudes comportamientos diferentes. Los factores asociados a la física del ambiente son comportamientos emergentes de acuerdo a las diferentes masas de los agentes que participan en la simulación, por ejemplo objetos que se pueden o no atravesar y hacen que los agentes modifiquen sus rutas o por cuando un agente con más masa desplaza a uno con menor masa. Los agentes están provistos de sentidos que les permiten re-calcular sus trayectorias para evitar colisionar con objetos dentro del ambiente y con otros agentes y utilizando las fórmulas propuestas por Samson cada agente toma velocidades diferentes al momento de desplazarse por el ambiente. Sin embargo es posible obtener velocidades muy similares en agentes que tengan valores en sus edades, pesos y alturas similares.

Los agentes virtuales se desplazan libremente en el ambiente virtual y determinan la dirección inicial hacia donde se desplazan aleatoriamente, esta dirección que toman se ve modificada solo cuando encuentran un obstáculo fijo o se encuentran a otro agente que impide continuar en la misma dirección, por lo que entonces el agente cambia aleatoriamente su ruta. Estos cambios aleatorios de dirección permiten que los agentes tomen distintas direcciones dentro del ambiente virtual, lo que permite observar cómo dos agentes que van hacia un mismo lugar tomen diferentes caminos. También agentes con mayor velocidad cambian de dirección más rápido que agentes que se desplazan a menor velocidad.

En la próxima sección se describe una técnica para categorizar distintas zonas del ambiente virtual y que permite generar un comportamiento de agrupación de los agentes que son dotados de un determinado rol.

3.2. Roles en agentes y su distribución usando diagramas de Voronoi

La geografía del ambiente es un factor importante en la generación de multitudes, los agentes virtuales que ocuparán el espacio deberán contar con información inicial que les permita reconocer su entorno y poder lograr sus objetivos, así mismo saber qué lugares son los que pueden visitar y los que deben de evitar. Para resolver esto, en nuestro caso hacemos uso de los diagramas de Voronoi que permiten dividir un área en regiones bien definidas.

3.2.1. Lugares de concentración de población (LCP)

Las multitudes suelen agruparse en distintos lugares del ambiente, estos lugares regularmente son destinos de las personas donde se detienen por un determinado tiempo, distintos lugares agrupan a distintas secciones de las multitudes dado que no todas las personas que conforman a una multitud tienen los mismos destinos. Para emular estos comportamientos el ambiente virtual tiene lugares que se han caracterizado por contener concentración de agentes que representan a distintos sectores de la población, a estos lugares los nombramos lugares de concentración de población (LCP).

Un LCP puede ser, por ejemplo, una escuela primaria (en México), donde es común encontrar más niños que adultos, entonces los agentes en la simulación con un rol de niño de entre 6 y 12 años se agrupan en las cercanías de dicho LCP, también es posible encontrar agentes adultos en este LCP pero serán minoría, mientras que en un LCP que represente un bar por ejemplo es más común encontrar a personas adultas, entonces los agentes adultos se van a agrupar en la región del LCP bar y otros agentes como por ejemplo los agentes con rol de niño pueden pasar por ese LCP pero no se van a quedar en la región. Para emular estos comportamientos de agrupación selectivos entre la población se requiere que los agentes con determinados roles se agrupen al acercarse a un LCP de su interés. Para ello se necesita comparar la distancia de cada agente contra el LCP más cercano y poder decidir si se agrupa en dicho LCP o debe continuar su camino

hasta encontrar un LCP de su interés, para que cada agente tenga conocimiento de cual es el LCP más cercano, para esto se utilizan los diagramas de Voronoi.

3.2.2. Vecino más cercano usando el diagrama de Voronoi

Una de las propiedades de los diagramas de Voronoi es que permiten identificar cuál es el vecino más cercano sin tener que re-calcular distancias cuando en nuestro caso un agente virtual se mueve de un lugar a otro, lo que permite obtener una simulación visualmente fluida. Por ejemplo si se tiene un plano con diferentes puntos los cuales van hacer los LCP (ver Figura 3.1).

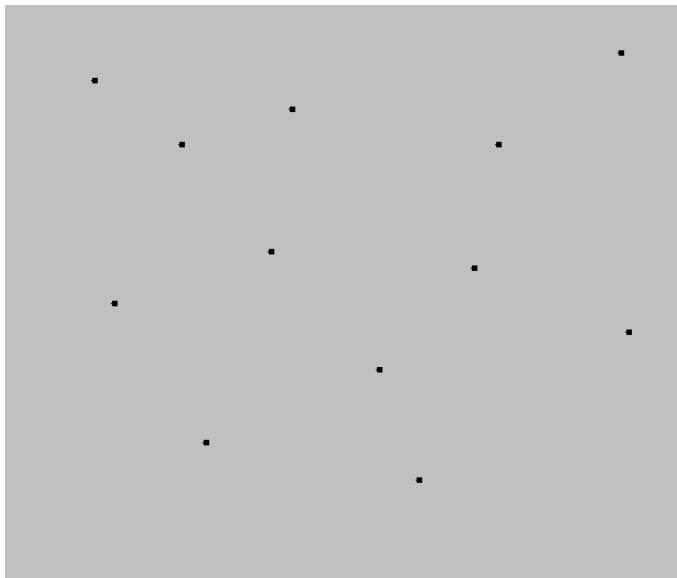


Figura 3.1: Plano con puntos (LCP) antes de calcular el diagrama de Voronoi

Al mismo plano si se le agrega otro punto que sera un agente en la simulación (ver Figura 3.2) y se requiere conocer cuál es su vecino mas cercano, una forma de resolverlo sería calcular las distancias entre todos los demás puntos para conocer cual es el más cercano, otra forma sería solo comparar la distancia de sus vecinos más cercanos.

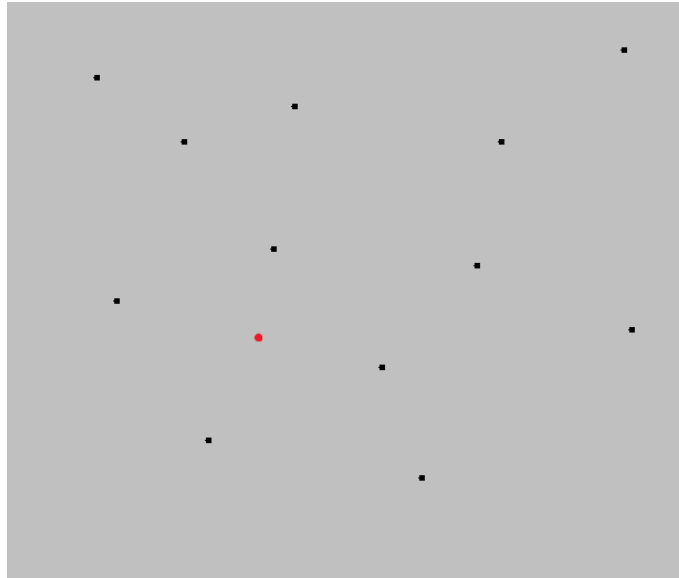


Figura 3.2: Plano con puntos (LCP), más un punto (agente virtual)

Sin embargo, si este punto está en constante movimiento y es lo que ocurrirá en la simulación cuando el agente virtual se mueva, se tendrían que estar recalculando las distancias para conocer cuál es el vecino más cercano y esto se tendría que hacer para cada uno de los agentes virtuales por lo que delimitaría la cantidad de agentes para obtener una simulación visualmente fluida, para solucionar esto se utiliza el diagrama de Voronoi el cual se calcula una vez (ver Figura 3.3)

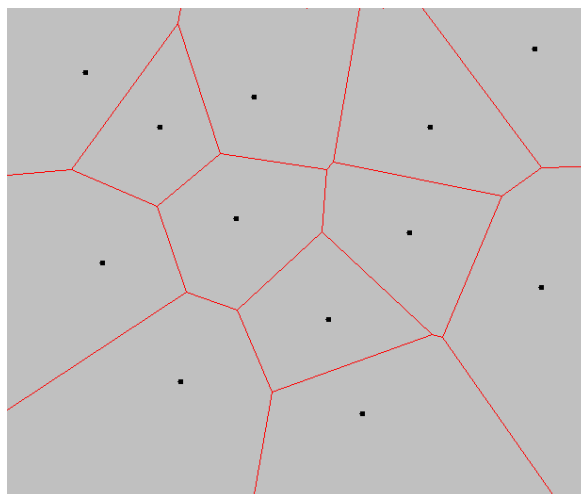


Figura 3.3: Diagrama de Voronoi del plano

Sustituyendo los puntos del plano por LCP de la simulación, una vez calculado

3. TÉCNICAS PROPUESTAS

el diagrama de Voronoi, los agentes que recorran el ambiente pueden conocer cuál es el LCP más cercano a ellos simplemente conociendo la región de Voronoi en la cual estén ubicados sin la necesidad de realizar más cálculos aun cuando estos se mueven de un lugar a otro.

Una vez que se conoce el diagrama de Voronoi del plano es sencillo saber cuál es el vecino más cercano incluso cuando el agente se desplaza por el plano, por ejemplo en la Figura 3.4, un agente (punto rojo) que se desplaza y que debe reconocer cuál es el LCP más cercano, únicamente compara la celda de Voronoi que pisa y en automático sabe cual es el LCP que se encuentra más próximo a ese agente, en este caso es el LCP parque.

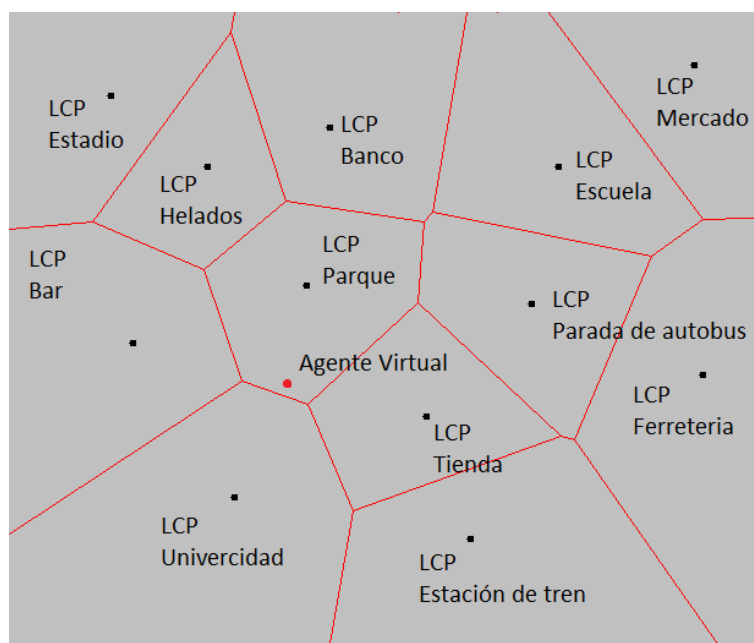


Figura 3.4: Diagrama de Voronoi del plano con los LCP y un agente

El caso de estudio hace uso del algoritmo Steven Fortune para generar el diagrama de Voronoi ya que dicho algoritmo [49] se ejecuta en $O(n \log n)$. Una vez generado el diagrama de Voronoi es utilizado para determinar la correcta distribución de la población dentro del ambiente virtual. Para los primeros resultados trabajaremos con una ciudadela simulada (ver Figura 3.5), donde los puntos p generados por Voronoi son los LCP. Un LCP puede ser una parada de autobús, una tienda comercial, etc.

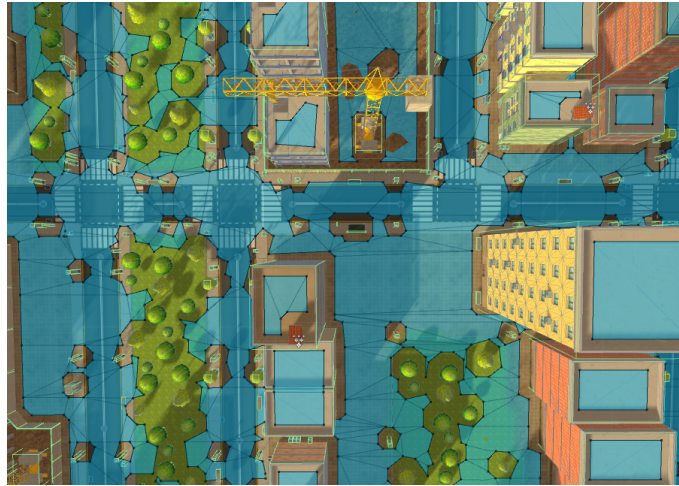


Figura 3.5: Ambiente virtual dividido en distintas regiones de Voronoi [10]

Para que distintos grupos de agentes virtuales se agrupen en los distintos LCP que pueden existir en el ambiente se asignó a los agentes virtuales distintos tipos de roles, dependiendo del rango de edad que tengan, los roles que se determinaron son: niño, joven, adulto y anciano, cada uno con sus variantes de hombre y mujer, los rangos se muestran en la Tabla 3.3.

Tabla 3.3: Roles y rangos de edades utilizados en los agentes virtuales

Rol	Rango de edad
Niño	5 - 14
Joven	15 - 32
Adulto	33 - 64
Anciano	65 - 99

Los rangos de edades se basaron en los rangos propuestos por la Organización Mundial de la Salud (OMS) en [68] que se pueden ver en la Tabla 3.4.

Tabla 3.4: Rangos de edades de la OMS

Rol	Rango de edad
Niño	5 - 11
Pubertad	12 - 14
Adolescencia	15 - 17
Jóvenes Adultos	18 - 32
Adulto	33 - 64
Anciano	65 - 99

3. TÉCNICAS PROPUESTAS

En esta investigación se agruparon los roles propuestos por la OMS de la siguiente manera; el rol de niño y pubertad de la OMS conforman el rol de niño, el rol de adolescente y jóvenes adultos conforman el rol de joven y en el caso de los adultos y acianos se mantienen los mismos rangos de edad, quedando los rangos como muestra la Tabla 3.3. El motivo para agrupar algunos roles es para utilizar únicamente 4 tipos de roles en lugar de los 6 propuestos por la OMS ya que los rangos de edades de niño, pubertad, adolescencia y jóvenes adultos tienen menos años de diferencia de los rangos en comparación con los de adulto y anciano.

Después de definir los roles de los agentes se redefinen los rangos posibles para las características de peso y altura, esto para evitar valores que se alejen de la realidad, porque en la vida real no se ha dado el caso de encontrar a un niño, por ejemplo de 6 años que mida 2 metros y pese 120 kilos, es por eso que los rangos para peso y altura se basan en la Tabla 3.5.

Tabla 3.5: Roles y rangos de altura y peso redefinidos

Rol	Rango de peso	Rango de altura
Niño	25 - 60 kg.	0.90 - 1.70 m.
Joven	40 - 120 kg.	1.30 - 2.00 m.
Adulto	50 - 120 kg.	1.50 - 2.00 m.
Anciano	50 - 100 kg.	1.20 - 1.90 m.

En el siguiente capítulo se describen algunas simulaciones así como los resultados que se obtuvieron de las mismas.

Resultados

El software utilizado para las simulaciones de los agentes fue Unity 5.3 de 64 bits, en una computadora con procesador intel i7 de 3.4GHz y 8 GB en RAM. En las primeras pruebas se representaron a los agentes con cubos en 3D, posteriormente se construyeron modelos de humanoides con algunas animaciones de diferentes tipos que remplazaron a los cubos en 3D. El ambiente virtual utilizado al principio es un terreno vacío con pasto y algunos muros, después se construyó una ciudad virtual de características básicas, cuenta con calles, bancas, alumbrado público, casas y edificios.

Tabla 4.1: Prueba de embotellamiento con 200 agentes virtuales

Número de prueba	Tiempo
1	10.7 s.
2	11 s.
3	11 s.
4	11.2 s.
5	11 s.

La primera prueba fue realizada con 200 agentes virtuales de un mismo tipo, el objetivo era probar su comportamiento en un espacio que se reducía, tipo cuello de botella y observar cómo se reagrupaban negociando los espacios con los demás agentes así como llegar a un destino o meta. La Figura 4.1 muestra la posición de salida y el cuello de botella que recorren los agentes, la prueba se ejecutó 5 veces y el promedio en tiempo que a los agentes virtuales les tomó llegar a su destino fue de 11 segundos (ver Tabla 4.1). Las variaciones de tiempo surgen porque los agentes no siempre elegían la misma ruta, cuando se reducía el espacio los agentes para no chocar entre sí cambiaban sus rutas para esquivarse, en cada simulación de diferente manera. Las 5 pruebas fueron suficientes para observar

4. RESULTADOS

que en cada simulación los agentes se comportaban diferente, como prueba se tiene la variación de tiempo entre cada simulación.

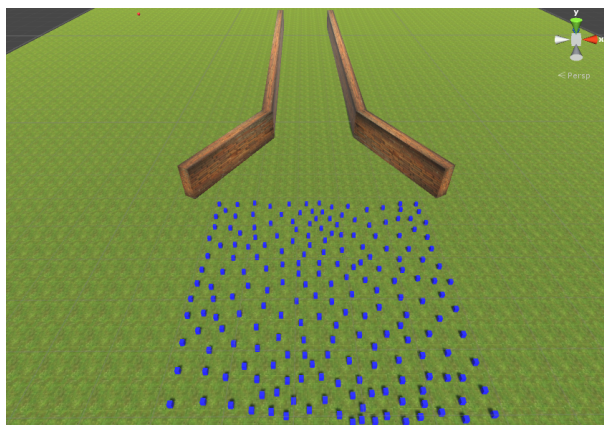


Figura 4.1: Desplazamiento de agentes virtuales en un espacio que se reduce

En la segunda prueba realizada con los mismos 200 agentes virtuales se colocaron obstáculos que dificultaban la ruta más corta a su objetivo por lo que los agentes tuvieron que buscar la ruta o rutas para llegar a su destino. Lo que se buscó probar como los agentes modificarían sus trayectorias evitando colisionar entre ellos, así como alcanzar su objetivo final. Se realizaron 5 ejecuciones (ver Tabla 4.2). La prueba que más tardó fue de un minuto con 8 segundos, debido a que cada agente virtual puede elegir un movimiento de dirección diferente. A algunos agentes les tomó más del doble del tiempo en llegar al objetivo en comparación con los primeros en llegar al objetivo. La Figura 4.2 muestra la posición de salida y los obstáculos de los agentes.

Tabla 4.2: Prueba de búsqueda de ruta para llegar al objetivo

Prueba	Primer agente en terminar	Último agente en terminar
1	20 s.	59 s.
2	27 s.	65 s.
3	25 s.	60 s.
4	30 s.	68 s.
5	26 s.	63 s.

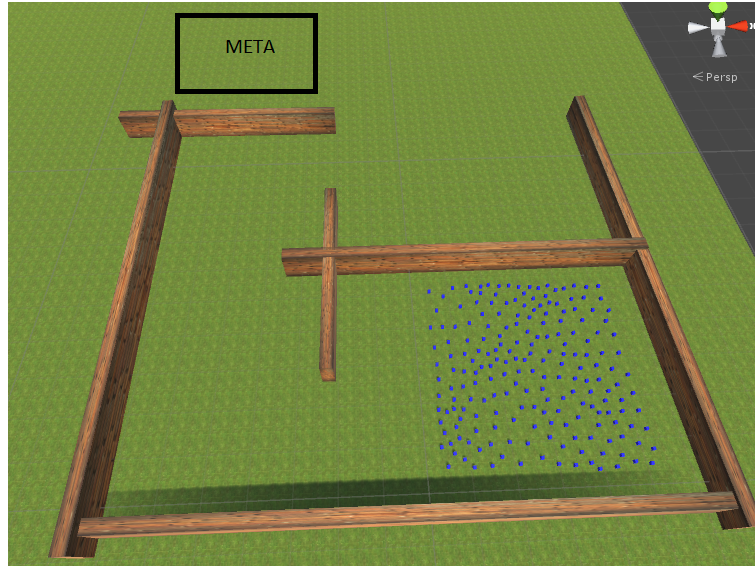


Figura 4.2: Agentes virtuales con obstáculos

Después se realizó una prueba con dos grupos de agentes virtuales, cada grupo con 200 agentes y en esta ocasión tenían diferentes masas, alturas y edades por lo tanto distintas velocidades y comportamientos de desvíos de rutas para evitar colisiones. La prueba consistió en que los dos grupos intercambiasen posiciones, la prueba se ejecutó 5 veces y en todas se distinguió como los agentes cambiaban sus direcciones para no chocar de frente con los agentes del otro grupo.

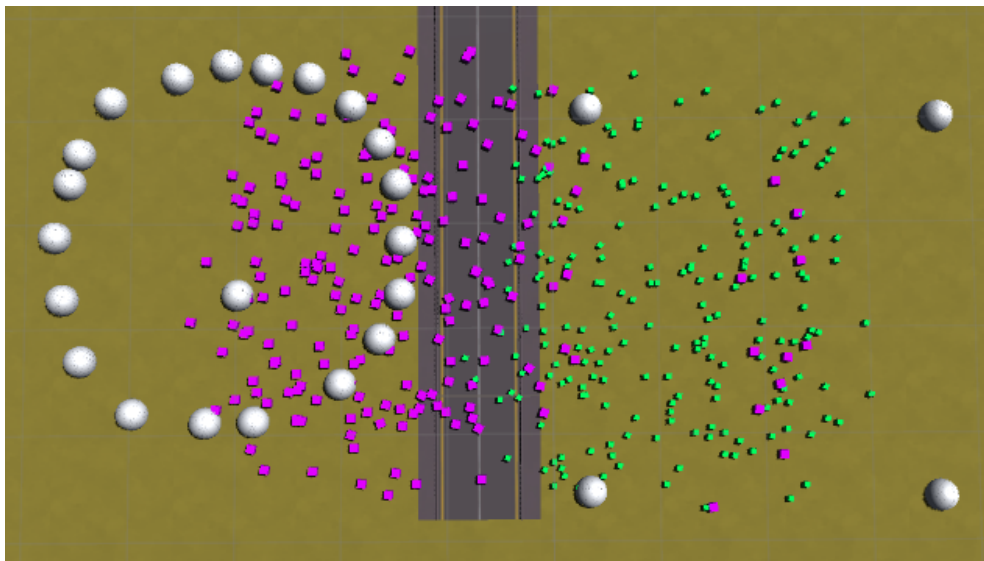


Figura 4.3: Multitudes intercambiando posiciones [10]

4. RESULTADOS

En las manifestaciones que involucran multitudes, estas se agrupan en pequeños grupos y se van uniendo a otros para llegar al objetivo final. Estos pequeños grupos deben sortear diversos obstáculos para alcanzar al contingente mayor, en la siguiente prueba lo que se hizo fue integrar dos grupos de multitudes con diferente patrón y observar como se realiza la integración en un solo grupo y qué patrón se obtiene como resultado. En la Figura 4.3, en el momento en que comienzan el desplazamiento los agentes rompen el patrón inicial y una vez que se forma el nuevo grupo se mantiene un patrón de forma cuadrada cuando se trata de desplazamiento, a la mitad de la prueba se produce el entrelazamiento de los grupos donde los agentes realizaban cambios de rutas para evitar chocar.

Uno de los aspectos importantes de las multitudes es la forma en cómo éstas se agrupan [18]. Los expertos en comportamiento humano han determinado ciertos patrones entre los que destacan la agrupación cuadrada, circular, delta y ovoide. El siguiente resultado se obtuvo al forzar a los agentes a agruparse en estos patrones, sin que exista una orden directa, esto lo hacen de acuerdo a la negociación de ocupar un espacio, sabiendo que la tarea se completa en el momento en que todos los agentes logren el patrón. La Figura 4.4 muestra los patrones obtenidos por los agentes, es importante resaltar que la distancia entre agentes no está dada, en caso contrario el resultado final sería una formación tipo militar (alineada), lo que no sucede en el comportamiento de una población real de civiles. El patrón delta se forma en presencia de un agente líder a seguir.

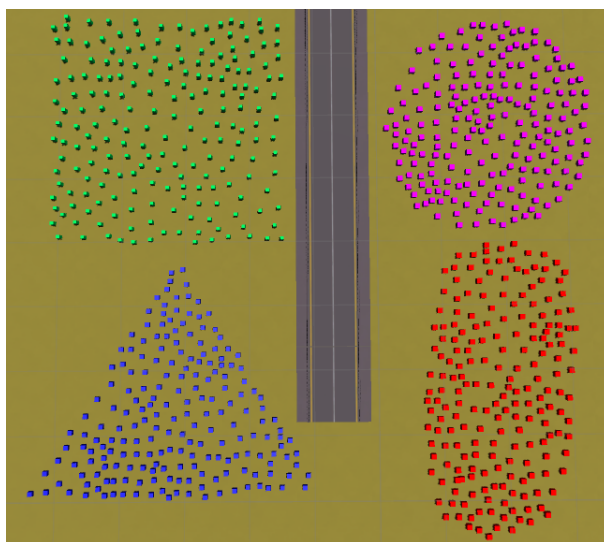


Figura 4.4: Patrones de multitudes obtenidas por los agentes [10]

Las siguientes pruebas se realizaron en una ciudad donde es posible distinguir comportamientos heterogéneos, los agentes son identificados por colores para cla-

sificarlos en 4 tipos de roles ya definidos dentro del ambiente. Los de color verde representan a la población de niños, los de color azul representan a jóvenes, los rojos representan a adultos, los violeta representan a los ancianos. La Figura 4.5 muestra la representación de los agentes poblando el medio ambiente virtual.

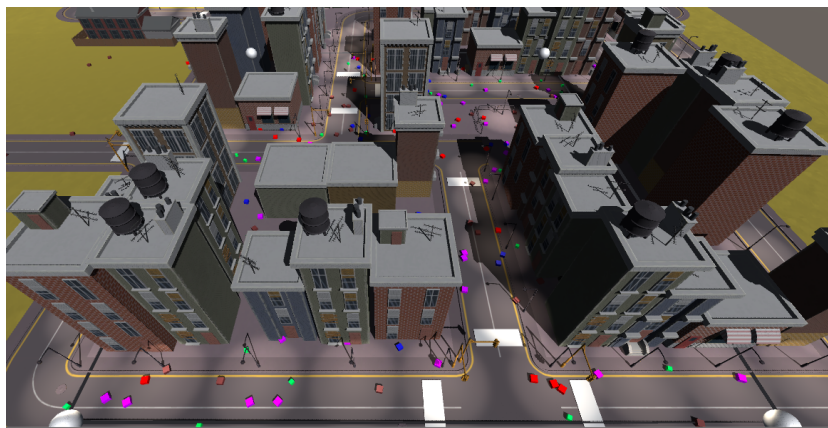


Figura 4.5: Agentes virtuales heterogéneos

En estas pruebas ya se implementan las características como peso, altura, edad y sexo en los agentes lo que permite usar las fórmulas de velocidad propuestas por Samson. Además ya se implementan algunos LCP, para que determinados agentes se vean atraídos a diferentes LCP y que los agentes tengan la posibilidad de tomar diferentes caminos ocupando todo el espacio geográfico del ambiente permitiendo lograr comportamientos heterogéneos.



Figura 4.6: Agentes virtuales desplazándose por la ciudad buscando su LCP de interés [10]

4. RESULTADOS

En la Figura 4.6 se muestra la simulación de los agentes desplazándose por las calles de la ciudad, rumbo a su LCP objetivo, se observa cómo los agentes toman en cuenta al resto de los agentes dentro del ambiente evitando colisionar con otros. Se observa cómo se acumuló una masa importante de agentes intentado acceder al recurso, lo que produce saturación en determinadas zonas.

Los LCP permiten concentrar agentes de un mismo tipo, sin embargo es válido encontrar agentes de otro tipo en la misma zona, esto es debido a las rutas que toman los agentes y son comportamientos similares a los vistos en la realidad. Finalmente la emergencia de comportamientos como la agrupación de multitudes resulta de la necesidad de acceder a un recurso, tal como sucede en la realidad. Estas pruebas tomaron un tiempo de 4 minutos aproximadamente, y los agentes se agrupaban en aproximadamente entre uno y dos minutos pero nunca se dejaban de mover ya que continuaban su movimiento dentro de la zona de su LCP.

En estas pruebas se utilizó el mismo número de agentes para cada rol, siendo en total 8 roles diferentes (ver Tabla 4.3) dando un total de 800 agentes.

Tabla 4.3: Roles y número de agentes virtuales por cada tipo de rol

Rol	Género	Número de agentes
Niño	Masculino	100
Niña	Femenino	100
Joven	Masculino	100
Joven	Femenino	100
Adulto	Masculino	100
Adulta	Femenino	100
Anciano	Masculino	100
Anciana	Femenino	100

Posteriormente se utilizó la herramienta online AUTODESK CHARACTER GENERATOR, ya que permite generar modelos de humanos completos y permite personalizarlos de forma gratuita y todo desde su web service en línea, una vez contruidos los modelos permite descargarlos en el formato compatible con Unity, para las simulaciones se crearon 8 modelos de criaturas virtuales para representar a los 8 roles de agentes virtuales que se habían creado, en la Figura 4.7 se observan los modelos creados.



Figura 4.7: Criaturas virtuales para representar a los agentes virtuales

En las primeras pruebas utilizando las criatura virtuales el rendimiento de la simulación disminuyó considerablemente ya que en las anteriores simulaciones se mantenía la simulación por los 30 cuadros por segundo, con las criaturas virtuales con solo una animación (caminar) y disminuyendo el número de agentes virtuales a 200 en total, la simulación se ejecutó a 12 cuadros por segundo, por lo que la simulación no se veía fluida, esto porque ahora el motor gráfico debía procesar los movimientos de los agentes al caminar así como también debía representar las sombras de los agentes en forma de siluetas correspondientes a las criaturas virtuales. En la Figura 4.8 se observan en el ambiente la integración de las criaturas virtuales que representan a los humanos en una ciudadela reemplazando a los cubos utilizados en las simulaciones anteriores.

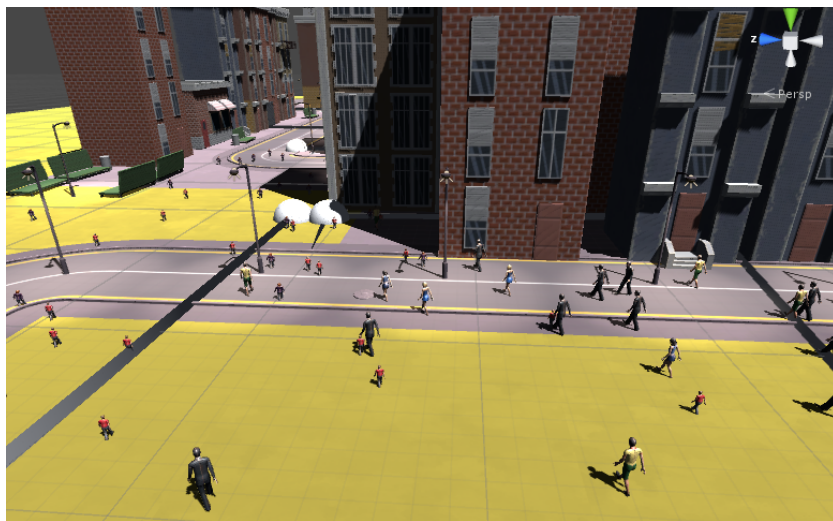


Figura 4.8: Agentes virtuales caminando por la ciudad

4. RESULTADOS

Representar a los agentes virtuales con cubos permitió simulaciones con un mayor número de agentes y un mayor número de cuadros por segundo, debido a que el procesamiento gráfico de cada cubo es menor en comparación a los modelos de humanos y el tiempo en renderizar la simulación también era menor, sin embargo los resultados con los modelos de humanos le daba una mejor representación visual, dado que la investigación se enfoca en generar una multitud de agentes que representen personas y aunque para esto se disminuya la cantidad de agentes virtuales se decidió continuar las simulaciones con los agentes representados por modelos de humanos.

Para optimizar la ejecución de la simulación en primera instancia se bajó la calidad a las sombras generadas por el motor gráfico, ya que el procesamiento de sombras generado por los entornos de luz del ambiente virtual utiliza el procesamiento máximo de la tarjeta de gráficos que tiene la computadora con la que se realizó la simulación y por ende al bajar la calidad de las sombras se requiere un menor procesamiento de gráficos.

Se lograron mejores resultados en la fluidez de visualización en la simulación reduciendo cálculos en la parte gráfica. Si bien para nuestro objetivo permitía observar a los agentes desplazarse por el ambiente, es necesario poder realizar este tipo de simulaciones con ordenadores que tengan prestaciones tanto en procesamiento como en visualización.

Se decidió quitar las sombras por completo a pesar de que esto se aleja de la realidad puesto que no interfería con los comportamientos heterogéneos generados ni con las agrupaciones de agentes que se formaban durante la simulación y si es un objetivo obtener una simulación fluida sin retrasos en la visualización.



Figura 4.9: Agentes virtuales sin sombras caminando por la ciudad

Los resultados tuvieron una mejoría en los cuadros por segundo, ya que subie-

ron hasta los 28 sin las sombras, la simulación se veía fluida, en la Figura 4.9 se ven algunos agentes virtuales sin las sombras recorriendo la ciudad, ya no fue necesario disminuir el número de agentes virtuales después de esta modificación.

En la Figura 4.10 se pueden ver algunos agentes agrupados en las cercanías de un LCP que en este caso representa una tienda de verduras que atrae más a los agentes con el rol de adultos y ancianos, aunque también es posible encontrar agentes con rol de niños en esa región, en la imagen no se encuentran agentes con el rol de niños o niñas porque no son atraídos por este LCP en específico.



Figura 4.10: Agentes virtuales cerca de un LCP de su interés

Conclusiones y trabajo futuro

En este capítulo se describen las conclusiones obtenidas en la investigación y se proponen algunas actividades que se pueden adicionar a la investigación como trabajo futuro.

5.1. Conclusiones

La generación de multitudes es un problema complejo a resolver cuando se aborda el concepto de heterogéneo, esto se debe a la dificultad de poder dotar a los agentes de características individuales, lo que no sucede en las simulaciones actuales para generar multitudes. El uso de diagramas de Voronoi permite generar zonas dentro del ambiente que pueden ser etiquetadas como un lugar de concentración donde los agentes pueden acceder a un recurso y que forma parte de su base de conocimientos. La inteligencia de los agentes se basa en la posibilidad de verificar las condiciones del ambiente y consultar, mediante el paso de mensajes, la ubicación dentro de su espacio. Comportamientos como agrupación, evasión, colisión, son determinados por técnicas de tipo *steering*.

El uso de las fórmulas propuestas por Samson permiten observar en la simulación desplazamientos en los agentes de forma variada lo que asemeja a lo observado en la realidad.

Los lugares de concentración (LCP) permiten atraer a los agentes por la necesidad de acceder al recurso que se encuentra como objetivo en su base de conocimiento lo que permite en determinado momento tener una concentración de agentes virtuales en un LCP específico.

La identificación de LCP permite observar concentración heterogénea de agentes de acuerdo a un contexto, la técnica de LCP puede ser utilizada en otros contextos, por ejemplo un parque puede tener distintos LCP internos, pudiendo

configurar el área de los columpios y resbaladillas para que se agrupen agentes virtuales con roles de niños y niñas, mientras que en el área de las bancas se pueden agrupar los agentes virtuales con roles de adultos, por ejemplo. La técnica puede ser utilizada para los problemas donde se puedan categorizar diversas áreas y determinados grupos de objetos que se tengan que agrupar en ellas.

La implementación de las técnicas propuestas fue codificada en scripts que fueron agregados de forma modular a los agentes virtuales, por lo que es posible agregar nuevos scripts con nuevos comportamientos para los agentes virtuales sin la necesidad de modificar los scripts actuales.

5.2. Trabajo futuro

Es necesario poder incorporar en la base de conocimientos de los agentes un tipo de personalidad que permita observar comportamientos lo más apegados a los observados en los individuos y no solamente hacer uso de técnicas aisladas. A manera de lista presentamos los trabajos que deben ser realizados para poder obtener comportamientos heterogéneos en agentes virtuales.

- Agregar nuevos comportamientos a los agentes virtuales.
- Ampliar las animaciones de la criaturas virtuales.
- Construir nuevos ambientes virtuales.
- Construir un generador de LCP automático editable.
- Agregar roles y criaturas virtuales a los agentes que representen animales.
- Agregar expresiones faciales a las criaturas virtuales para retroalimentar al usuario en este tipo de simulaciones.

Publicaciones

Durante la investigación se participó en el 4th International Symposium on Language and Knowledge Engineering (LKE), by the Faculty of Computer Science at the Benemérita Universidad Autónoma de Puebla (BUAP) el 3 y 4 de Noviembre del 2016 en Puebla, México. Lo que produjo la publicación del artículo:

Fernando Rebollar Castelan, Marco A. Ramos Corchado, Vianney Muñoz Jiménez and Félix F. Ramos Corchado. Generación de multitudes virtuales heterogéneas basadas en patrones de agrupación de comportamiento humano. *Research in Computing Science Journal*, ISSN 1870-4069, vol 124, 2016.

El artículo mencionado se expuso en el mismo congreso el 4 de Noviembre del 2016.

Generación de multitudes virtuales heterogéneas basadas en patrones de agrupación de comportamiento humano

Fernando Rebollar Castelan¹, Marco A. Ramos Corchado¹, Vianney Muñoz
Jiménez¹, and Félix F. Ramos Corchado²

¹ Universidad Autónoma del Estado de México (UAEMex),
Facultad de Ingeniería

² Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional
(CINVESTAV) Unidad Guadalajara
fc@live.com.mx, marco.corchado@gmail.com, vmunozj@uaemex.mx,
framos@gdl.cinvestav.mx

Resumen. Las ciencias computacionales en conjunto con otras áreas del conocimiento participan en el estudio del comportamiento de masas de individuos, intentando predecir y anticipar situaciones que puedan presentarse, a través de simulaciones de multitudes que permitan una experimentación a bajo costo y que aporten datos a las organizaciones para predecir y anticipar distintas eventualidades. Para ello, es necesario realizar simulaciones con multitudes virtuales que se comporten lo más parecido a la realidad. En este artículo se presenta una forma de generar multitudes virtuales con comportamientos heterogéneos, de tal forma que los individuos que conforman la multitud tengan distintos comportamientos. Se propone una técnica de agrupamiento en distintas regiones del área a poblar utilizando diagramas de Voronoi, posibilitando la caracterización de zonas donde existe una concentración masiva de personas en un espacio específico como lo son: plazas, centros comerciales, aeropuertos, escuelas, etc.

Palabras Clave. Inteligencia artificial, agentes, multitudes heterogéneas.

Generation of Heterogeneous Virtual Crowds based on Patterns of Grouping of Human Behavior

Abstract. Computer science with other areas of knowledge is involved in the study of the individual's behavior of the crowd, trying to predict and anticipate situations that may arise. In crowd simulations, we can see different experimentations that provide the different behavior on specific conditions or eventualities not controlled. In this paper, we present one technique based on Voronoi diagrams to defined the local concentration of people (LCP) that permitted create the heterogeneous crowd. LCP identified the principal place of environment where to exist the possibility of a crowd. The Voronoi diagrams help to agents to find the particular location to access a resource to complete the main goal. The first results showed the way of how the agents can crowd like the circles, delta, and oval grouping.

Keywords. Artificial intelligence, agents, heterogeneous crowds.

1. Introducción

El estudio de la concentración de individuos en lugares públicos como plazas, centros comerciales, parques, jardines, etc., es un campo de estudio abierto en las diferentes disciplinas de las ciencias. Conllevando a la necesidad de disponer de sistemas que permitan pronosticar y predecir eventualidades en situaciones no controladas, como es en el caso de un terremoto. De ahí que, las ciencias computacionales como la inteligencia artificial investigan cómo replicar el comportamiento humano virtualmente, para obtener simulaciones que sean lo más pegado a la realidad, apoyándose de las áreas sociales, psicológicas, neurocientíficas, entre otras, con la finalidad de empujar estas teorías en el área de la inteligencia artificial.

El principal propósito de estudiar y simular multitudes virtuales es representar con precisión grupos de individuos autónomos llamados agentes virtuales que atienden a las mismas reglas en entornos cercanos a aquellos encontrados en la vida real. Para anticipar contingencias y atenderlas adecuadamente antes de que sucedan es necesario estudiar el comportamiento de las masas de individuos en actividades cotidianas, permitiendo la construcción de la infraestructura requerida de acuerdo a las necesidades de los individuos que hacen uso de ella, por ejemplo, las plazas, los aeropuertos, las escuelas, el transporte público, etc. A esta infraestructura se le conoce como ciudades inteligentes.

Grupos de investigación alrededor del mundo se han sumado al problema de la simulación de multitudes virtuales con grandes logros desde el realismo obtenido hasta el comportamiento embebido en los agentes virtuales. Sin embargo, sus simulaciones carecen de heterogeneidad en la población ya que sus estudios se enfocan en un sólo modelo que replican infinitamente en el medio ambiente virtual, esto resulta menos costoso computacionalmente comparado a la generación diversos modelos de acuerdo a la heterogeneidad de la población e implica que el comportamiento observado dentro de la simulación sea idéntico para todos los agentes virtuales [1].

En la realidad no existe un número determinado de individuos que constituyen una multitud, no obstante, el estudio de las multitudes se basa en el mayor número de individuos concentrados en un espacio observando su comportamiento en situaciones de estrés, manteniendo la meta de poder anticipar contingencias en tiempo real [2]. Las aproximaciones computacionales para la generación de multitudes son centradas en las reglas propias del ambiente físico y el comportamiento está basado en la interacción del ambiente y las reglas físicas del mismo.

Uno de los problemas que se tienen con la generación de multitudes es la diversidad visual para representar la heterogeneidad en las multitudes. Así como la planificación que deberán de realizar los agentes dentro del ambiente virtual para el logro de sus objetivos [3].

Para reproducir comportamientos individuales y grupales de forma exitosa dos problemas fundamentales deben abordarse: la planificación y la toma de decisiones. La planificación dotará a los agentes para poder observar el medio ambiente y poder decidir sobre las condiciones del mismo, por ejemplo la evasión

de colisiones. Los métodos principales para la evasión de colisiones se basan en fuerzas sociales [4] mientras que la velocidad es manejada de manera recíproca para todos los agentes. Un algoritmo simple para evitar colisiones compara la posición de cada agente con los demás, sin embargo conforme el número de agentes (denotado por N) crece, la complejidad del algoritmo es $O(N^2)$. Dicha complejidad representa un problema cuando se requiere simular grandes multitudes e imposibilita la obtención de una simulación en tiempo real. Para que una simulación de multitudes sea precisa es necesario que reproduzca comportamientos humanos individuales y grupales, además los algoritmos que sintetizan estos comportamientos deben estar optimizados para trabajar en tiempo real [5].

El comportamiento colectivo en los humanos se ha estudiado desde principios del siglo XIX con la finalidad de observar las acciones que las personas realizan cuando se presentan diferentes circunstancias como celebraciones, manifestaciones, simulacros o incluso cuando ocurren fenómenos naturales como incendios o terremotos [6]. Las ciencias computacionales realizan esfuerzos para simular de manera virtual multitudes con el objetivo de reproducir comportamientos similares a los de los seres humanos con la finalidad de estudiar eventualidades en las multitudes [1].

Las simulaciones de multitudes son realizadas a partir de diferentes enfoques dependiendo del ambiente virtual que se desea poblar, por ejemplo: la industria del entretenimiento (videojuegos, películas, realidad virtual, etc.). Este tipo de simulaciones ha tomado importancia en el uso de los llamados *juegos serios* [7]. Estos últimos requieren ser poblados con agentes que permitan al usuario contar con una retroalimentación o ser asistido en tareas inmersas en el ambiente virtual.



Fig. 1: Multitud de personas en avenidas publicas

Este tipo de simulaciones permite a las diferentes organizaciones garantizar la integridad de los individuos aglomerados en espacios públicos, como se ilustra en la Figura 1. Las organizaciones hacen uso de las simulaciones basadas en este tipo con el propósito de prevenir situaciones no controladas en el entorno [8].

2. Trabajos Relacionados

El estudio de las multitudes es de suma importancia y se puede observar ante la presencia de fenómenos naturales, por ejemplo los cardumen de sardinas, en donde las especies jóvenes se encuentran en el centro para protegerse de los predadores y preservar la especie. En 1987 Craig Reynolds presentó uno de los primeros trabajos sobre la creación de multitudes basados en el comportamiento de las aves y la manera de cómo se agrupan en vuelo. Las investigaciones de Reynolds encaminan a la generación de multitudes basadas en humanoides siguiendo tres reglas simples: separación, alineación y cohesión. Según Reynolds un ave es consciente de tres elementos durante el vuelo: conocimiento de sí mismo, vecinos cercanos, y un líder a seguir [9].

Los trabajos de Reynolds funcionan en agentes que tienen un mismo comportamiento como son las aves y otro tipo de animales. Sin embargo, las multitudes basadas en humanos es más compleja debido a la personalidad de cada uno de los individuos que participan en una multitud, donde diversos comportamientos deben de ser considerados para que la simulación se apegue más a la realidad.

Es importante considerar en una simulación de multitud un comportamiento reactivo, es decir asociar todas aquellas acciones que son desencadenadas mediante un evento, en tanto no se registre ningún estímulo del ambiente o incluso de algún otro agente involucrado en la simulación. Los agentes que pertenezcan a la categoría reactiva no tienen ninguna razón para reaccionar [1]. Erik Millan genera máquinas de estado finitas desde archivos XML y las guarda en imágenes que los agentes pueden consultar [2]. Pelechano y Badler han combinado reglas de percepción y comportamientos reactivos para dirigir agentes en entornos virtuales [10]. Si los comportamientos reactivos son implementados correctamente pueden tener como consecuencia comportamientos emergentes que producen simulaciones más realistas.

Las simulaciones basadas en reglas no requieren de un razonamiento complejo, esto es debido a que el medio ambiente es el responsable de seleccionar la mejor acción que los agentes deben seguir en las diferentes situaciones que se les presentan [11]. Sin embargo, esto se aleja a lo que sucede en un entorno real porque el ambiente no controla las acciones de los peatones, sólo los limita. Kapadia [12], combinó predicciones de espacio-tiempo, comportamientos reactivos, y movimientos de dirección en plataformas dedicadas a la simulación de multitudes, lo que da libertad a los agentes de tomar sus decisiones de desplazamiento por ellos mismos.

Lograr simulaciones de multitudes lo más parecidas a la realidad, requiere de resolver el problema de coaliciones que medido computacionalmente es de $O(N^2)$, donde N es el número de agentes que participaran en la simulación [5]. Esto significa que cada agente tiene que consultar a todos los otros agentes por su posición y otra información importante que le permita calcular su dirección y velocidad con respecto a la de sus vecinos. Esto reduce la complejidad de las búsquedas de proximidad, permitiendo simulaciones de grandes multitudes.

Para reducir la complejidad en el paso de mensajes entre los agentes se utilizan estructuras jerárquicas, como los *octrees*, en donde el espacio se subdivide

en varias regiones que contienen agentes donde los integrantes de una región conocen solo a los agentes que estén dentro de la misma región [13]. Mejoras a la estructura *octrees*, utilizan *árboles kd* para que al consultar a los vecinos más cercanos sea un proceso eficiente. Bleiweiss en [14] presentó una implementación en paralelo de la biblioteca popular para evasión de colisiones, obteniendo un aumento de velocidad de 4.8X en comparación con la implementación original. Bleiweiss cambió el método de búsquedas de proximidad de un *árboles kd* a un método basado en tablas *hash* con el fin de mejorar el rendimiento en la Unidad de Procesamiento Gráfico (GPU).

En 2004, Cheney [15] propone una técnica para la representación y el diseño de campos de velocidad, usando autómatas celulares útiles para crear movimientos de flujo, que son seguidos por los peatones, con el fin de moverse a través de un entorno, ver la Figura 2. En 2011, Zhang [16] presenta un modelo en el que las celdas de un autómata celular representan posiciones discretas en el espacio, utilizadas por los peatones para moverse al cambiar de una celda a otra generando simulaciones más realistas.

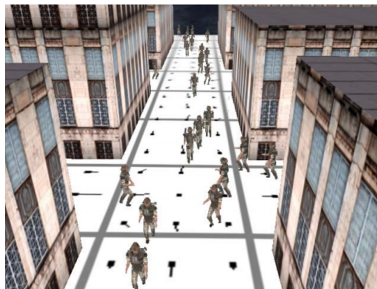


Fig. 2: Simulación de Cheney usando autómatas celulares [15]

Guy en 2010, presenta un método que calcula trayectorias reduciendo al mínimo el esfuerzo que los agentes necesitan llevar a cabo para llegar a su destino [17]. El algoritmo es capaz de evitar colisiones con otros agentes y los obstáculos, al tiempo que permite simulaciones en tiempo real [18].

Van den Berg mejora el algoritmo de RVO (Velocidades Recíprocas para Obstáculos) reduciendo el problema a un programa lineal de baja dimensión [19], y con ello son capaces de simular multitudes de miles de personajes. En la Figura 3 se observa una prueba con 1,000 agentes en tiempo real.

D. Thalmann [20], remarca que las principales razones que dificultan el uso de multitudes virtuales es precisamente el dominio de tiempo real y las altas exigencias al CPU, así como los altos costos de la producción de contenidos. Thalmann propone algoritmos para optimizar el hardware haciendo posible mostrar escenas virtuales en 3D con miles de entidades individuales animadas que anteriormente no era posible, su simulador permite crear miles de agentes donde la multitud se mueve de un lugar a otro en tiempo real (ver Figura 4).

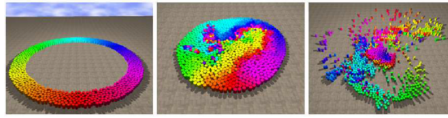


Fig. 3: Los agentes se mueven hacia la posición opuesta de donde comienzan en el círculo

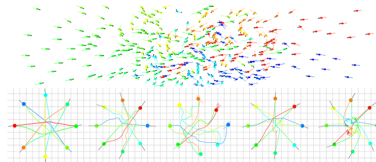


Fig. 4: Simulación en tiempo real de Thalmann [21]

3. Generación de comportamientos heterogéneos

Un problema a abordar en la generación de multitudes es la carencia de diversidad que la conforman, debido a que sólo se utiliza un modelo del agente que se replica de manera discriminada, cambiando sólo el color de la ropa o cabello. También podemos observar la falta de autonomía en los agentes provocando un solo comportamiento lo cual reduce el realismo en la simulación.

Nuestra propuesta se basa en la generación de multitudes heterogéneas a través de diversos modelos que conforman la multitud sin descuidar los comportamientos asociados a cada agente. Por ejemplo, sí en la escena se observa un anciano este debe tener el comportamiento de un anciano.

3.1. El comportamiento humano

El comportamiento de cada persona en la vida real es único, el cual esta determinado por un sin fin de factores que lo van determinando a lo largo del tiempo y de experiencias vividas por cada persona. También lo delimitan aspectos fisiológicos y capacidades que impiden poder realizar algunas actividades para determinados sectores de la población.

Lo ideal en las simulaciones de multitudes es que cada agente se comportara de manera diferente tal y como pasa en la vida real. Sin embargo, este proceso tomaría demasiados recursos del sistema para que cada agente virtual contara con un comportamiento individual impidiendo su simulación en tiempo real.

Samson en [22] determinó la velocidad de desplazamiento, cadencia de los pasos y longitud de zancada de las personas, considerando parámetros como la edad, el peso y la altura de las personas. Para ello, analizó a 118 mujeres y 121 hombres en un rango de entre los 19 a 90 años de edad, los cuales caminaban a su velocidad preferida como lo harían normalmente a través de una pasarela de 12 metros. Samson propone formulas para calcular la velocidad de desplazamiento de los individuos en actividades cotidianas. En nuestro caso de estudio

retomamos las ecuaciones propuestas por Samson para que los agentes virtuales reproduzcan estos desplazamientos con la característica que podemos distinguir hombres y mujeres dentro de la simulación. La Tabla 1 muestra la relación de velocidad desplazamiento existente entre hombres y mujeres.

Tabla 1: Ecuaciones de velocidades de desplazamiento en metros/segundos de los seres humanos de Samson [22]

Hombres
Velocidad = 1.460
Velocidad = -0.002 edad (*) + 1.582
Velocidad = -0.002 edad (*) + 0.442 altura (*) + 0.750
Velocidad = -0.001 edad (*) + 0.486 altura (*) - 0.001 peso (*) + 0.720
Mujeres
Velocidad = 1.420
Velocidad = -0.003 edad (*) + 1.552
Velocidad = -0.002 edad (*) + 0.618 altura (*) + 0.484
Velocidad = -0.001 edad (*) + 0.827 altura (*) - 0.003 peso (*) + 0.316

La posibilidad de dotar a los agentes de valores aleatorios en el desplazamiento, así como condicionantes de masa y altura permite obtener una simulación dentro de la multitud lo más cercana a la realidad. Los factores asociados a la física del ambiente son comportamientos emergentes de acuerdo a las diferentes masas de los agentes que participan en la simulación. Los agentes están provistos de sensores que les permiten recalcular sus trayectorias para evitar colisionar con objetos dentro de ambiente y con otros agentes. En un sistema multiagentes el ambiente se define como el conjunto $E = \{e, e', \dots\}$ donde E es el ambiente conformado por todos los posibles estados. $Ac = \{\alpha, \alpha', \dots\}$ representa todas las acciones permitidas dentro de E . Los agentes construyen su base de conocimientos a partir de $r : e_0 \xrightarrow{\alpha_0} e_1 \xrightarrow{\alpha_1} e_2 \xrightarrow{\alpha_2} e_3 \xrightarrow{\alpha_3} \dots \xrightarrow{\alpha_{u-1}} e_u$ que representan las acciones realizadas de un estado a otro dentro de E .

3.2. Distribución Geométrica Espacial del Ambiente

La geografía del ambiente es un factor importante en la generación de multitudes, los agentes virtuales que ocuparán el espacio deberán contar con información inicial que les permita reconocer su entorno y poder lograr sus objetivos, así mismo saber qué lugares son los que pueden visitar y los que deben de evitar. Para resolver esto, en nuestro caso hacemos uso de los diagramas de Voronoi que permiten dividir un área en regiones bien definidas.

Sea $P = \{p_1, p_2, \dots, p_n\}$ un conjunto de puntos en el plano, haciendo uso del diagrama de Voronoi es posible asignar a cada punto una región en el plano correspondiente a los puntos más cercanos, una región para cada $p_i \in P$, todos los puntos asignados a p_i en el conjunto de puntos P forman la región de Voronoi $V(p_i)$ [23]. Dado un conjunto de puntos en P y un punto de consulta q , es posible

determinar el punto más cercano a q en P , dado que la ubicación de q esta dentro de una región de Voronoi en un punto p_i la cual indica que dicho punto p_i es el más cercano al punto q .

$$V(p_i) = q \mid \|p_i q\| < \|p_j q\|, \forall j \neq i \quad (1)$$

Donde $\|pq\|$ es la distancia euclídea entre p y q .

Como primer aproximación podemos decir que dado un conjunto P de sitios (puntos) en el plano, su diagrama de Voronoi es la partición de ese plano en regiones (una región para cada sitio), tal que la región del sitio p contiene todos los puntos del plano que están más cerca de p que de cualquier otro sitio en P .

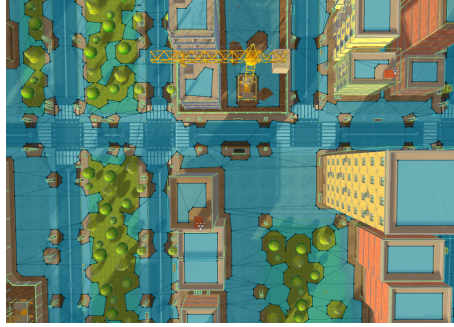


Fig. 5: Ubicación de LCP dentro del ambiente

Nuestro caso de estudio hace uso del algoritmo Steven Fortune para generar el diagrama de Voronoi ya que dicho algoritmo [24] se ejecuta en $O(n \log n)$ por lo que se puede notar que es un algoritmo en $O(n)$ considerado uno de los mejores actualmente. Una vez generado el diagrama de Voronoi es utilizado para determinar la correcta distribución de la población dentro de ambiente virtual, para los primeros resultados trabajaremos con una ciudadela simulada (ver Figura 5), donde los puntos p generados por Voronoi los llamaremos lugares de concentración de población (LCP). Un LCP puede ser una parada de autobús, una tienda comercial, etc. El uso de LCP nos permite conceptualizar los espacios dentro del ambiente y poder definir la concentración de individuos por regiones, por ejemplo si la región esta marcada como escuela primaria, dicha región deberá ser poblada en su mayoría de agentes que representen niños y niñas en un rango de edad entre los 6 y 12 años.

La posibilidad de contar con varios LCP nos permitirá poblar ambientes mucho más complejos y así poder observar y estudiar los comportamientos asociados y emergentes dentro de una población completamente heterogénea.

4. Resultados

El software utilizado para las simulaciones de los agentes fue Unity 5.3 de 64 bits, en una computadora con procesador intel i7 de 3.4GHz y 4 GB en RAM. Los primeros resultados obtenidos se basan en los desplazamientos que toman los agentes dentro del ambiente virtual en donde es posible distinguir desplazamientos heterogéneos, inicialmente los agentes están representados por cubos de diferentes tamaños y masas para dotarlos de características individuales.



Fig. 6: Agentes virtuales, desplazándose por la ciudad buscando su LCP de interés

Los agentes son identificados por colores para clasificarlos en 5 tipos de roles dentro del ambiente. Los de color verde representan a la población de niños entre 4 a 14 años de edad, los de color azul representan a jóvenes de 15 a 24 años, los rojos representan a adultos de 25 a 54 años de edad, los violeta representan a adultos mayores de 55 a 64 años y por último los de color café representan a los ancianos de más de 65 años. La Figura 6 muestra la representación de los agentes poblando el medio ambiente virtual.

La implementación de las ecuaciones de velocidad propuestas por Samson permiten observar desplazamientos de tipo natural. Además de la creación de los LCP, los agentes tienen la posibilidad de tomar diferentes caminos ocupando todo el espacio geográfico del ambiente permitiendo lograr comportamientos parecidos a los reales.

En la Figura 6 se muestra la simulación de los agentes desplazándose por las calles de la ciudad, rumbo a su LCP objetivo, se observa como los agentes toman en cuenta al resto de los agentes dentro del ambiente evitando colisionar con otros. Se puede ver así mismo como se comienza acumular una masa importante de agentes intentando acceder al recurso, lo que produce saturación en determinadas zonas. Los LCP permiten concentrar agentes de un mismo tipo, sin embargo es válido encontrar agentes de otro tipo en la misma zona, esto es debido a las rutas que toman los agentes y son comportamientos similares a los vistos en la realidad. Finalmente la emergencia de comportamientos como la agrupación de multitudes resulta de la necesidad de acceder a un recurso, tal como sucede en la realidad.

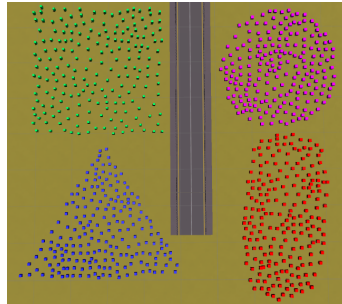


Fig. 7: Patrones de multitudes obtenidas por los agentes

Uno de los aspectos importantes de las multitudes es la forma en cómo estas se agrupan [25]. Los expertos en comportamiento humano han determinado ciertos patrones en los que destacan la agrupación cuadrada, circular, delta y ovoide. El siguiente resultado se obtuvo al forzar a los agentes a agruparse en estos patrones, sin que exista una orden directa, esto lo hacen de acuerdo a la negociación de ocupar un espacio, sabiendo que la tarea es completa en el momento en que todos los agentes logren el patrón. La Figura 7 muestra los patrones obtenidos por los agentes, es importante resaltar que la distancia entre agentes no está dada, en caso contrario el resultado final sería una formación tipo militar (alineada), lo que no sucede en el comportamiento de una población real de civiles.

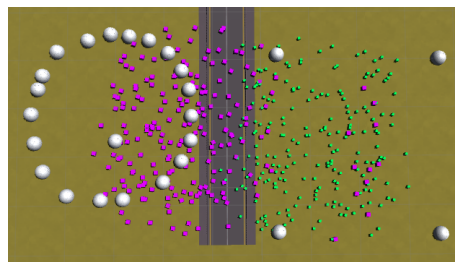


Fig. 8: Patrones de multitudes obtenidas por los agentes

En muchas de las manifestaciones que involucran multitudes estas se agrupan en pequeños grupos y se van uniendo a otros para llegar al objetivo final. Estos pequeños grupos deben sortear diversos obstáculos para alcanzar al contingente mayor, en nuestra siguiente prueba a manera de experimentación lo que hacemos es integrar dos grupos de multitudes con diferente patrón y observar como se realiza la integración en un solo grupo y que patrón se obtiene como resultado. Este comportamiento se observa en la Figura 8, en el momento en que comienzan el desplazamiento estos rompen el patrón inicial y una vez que se forma el nuevo grupo se mantiene un patrón de forma cuadrada cuando se trata

de desplazamiento y de forma circular u ovoide en el caso de espera. El patrón delta se forma en presencia de un agente líder a seguir.

En el vídeo ³ en la web se pueden observar dos experimentaciones en donde los agentes se desplazan a sus objetivos negociando espacios y calculando sus rutas, también en el mismo vídeo se puede observar la integración de las dos multitudes que se muestran en la Figura 8.

5. Conclusiones

El uso de las velocidades de desplazamiento de los seres humanos utilizado en las simulaciones de multitudes, permite observar comportamientos parecidos a los reales. La categorización de los individuos por edades genera comportamientos indistintos, lo que permite observar en la simulación comportamientos lo más parecidos a la realidad. Una de las aportaciones que se realizan en este trabajo es la creación de LCPs utilizando diagrama de Voronoi lo que nos permite generar concentraciones de individuos en un espacio al cual los agentes necesitan acceder como recurso, además de ver como los agentes realizan procesos de comunicación y negociación para el logro de sus objetivos individuales. La identificación de LCPs permite observar concentración heterogénea de agentes de acuerdo a un contexto, por ejemplo un parque deberá ser poblado en su mayoría por ancianos, niños y mujeres. Poder reproducir los patrones de las agrupaciones de las multitudes con los agentes virtuales nos permite estudiar los comportamientos como suceden en la vida real, permitiendo a las organizaciones evaluar las condiciones e implementar las políticas de posibles contingencias.

Referencias

1. De Gyves, O., Toledo, L., Rudomin, I.: Comportamientos en simulación de multitudes: revisión del estado del arte. *Research in Computer Science* (2013) 319–334
2. Millan, E., Hernández, B., Rudomin, I.: Large crowds of autonomous animated characters using fragment shaders and level of detail. *ShaderX5: Advanced Rendering Techniques* (2007) 501–510
3. Thalmann, D., Grillon, H., Maim, J., Yersin, B.: Challenges in crowd simulation. In: *2009 International Conference on CyberWorlds, IEEE* (2009) 1–12
4. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. *Physical review E* **51** (1995) 4282
5. Ruiz, S., Hernández, B.: Procesos de decisión de markov y microescenarios para navegación y evasión de colisiones para multitudes. *ResearchGate* (2014)
6. G, L.: *Psychologie des foules*. Alcan, Paris (1895)
7. Ritterfeld, U., Cody, M., Vorderer, P.: *Serious games: Mechanisms and effects*. Routledge (2009)
8. Zhong, Z., Ding, N., Wu, X., Xu, Y.: Crowd surveillance using markov random fields. In: *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on, IEEE* (2008) 1822–1828

³ <https://youtu.be/GaEpYtvak04>

9. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics* (1987)
10. Pelechano, N., Allbeck, J.M., Badler, N.I.: Controlling individual agents in high-density crowd simulation. In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, Eurographics Association (2007)* 99–108
11. Sun, L., Qin, W.: Simulation of crowd behaviors based on event reaction. In: *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on. Volume 2., IEEE (2011)* 163–167
12. Kapadia, M., Singh, S., Reinman, G., Faloutsos, P.: A behavior-authoring framework for multiactor simulations. *Computer Graphics and Applications, IEEE* **31** (2011) 45–55
13. Hadap, S., Eberle, D., Volino, P., Lin, M.C., Redon, S., Ericson, C.: Collision detection and proximity queries. In: *ACM SIGGRAPH 2004 Course Notes, ACM (2004)* 15
14. Bleiweiss, A.: Multi agent navigation on the gpu. In: *GDC09 Game Developers Conference. Volume 2009. (2009)*
15. Cheney, S.: Flow tiles. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA (2004)* 233–242
16. Zhang, S., Li, M., Li, F., Liu, A., Cai, D.: A simulation model of pedestrian flow based on geographical cellular automata. In: *Geoinformatics, 2011 19th International Conference on, IEEE (2011)* 1–5
17. Guy, S.J., Chhugani, J., Curtis, S., Dubey, P., Lin, M., Manocha, D.: Pedestrians: a least-effort approach to crowd simulation. In: *Proceedings of the 2010 ACM SIGGRAPH/Eurographics symposium on computer animation, Eurographics Association (2010)* 119–128
18. Paolo Fiorini, Z.S.: Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research* (1998) 760–772
19. Van Den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. In: *Robotics research. Springer (2011)* 3–19
20. Thalmann, D.: Populating virtual environments with crowds. In: *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications, ACM (2006)*
21. Pettré, J., Ciechomski, P.d.H., Maïm, J., Yersin, B., Laumond, J.P., Thalmann, D.: Real-time navigating crowds: scalable simulation and rendering. *Computer Animation and Virtual Worlds* **17** (2006) 445–455
22. Samson, M., Crowe, A., De Vreede, P., Dessens, J., Duursma, S., Verhaar, H.: Differences in gait parameters at a preferred walking speed in healthy subjects due to age, height and body weight. *Aging Clinical and Experimental Research* **13** (2001) 16–21
23. De Berg, M., Van Kreveld, M., Overmars, M., Schwarzkopf, O.C.: Computational geometry. In: *Computational geometry. Springer (2000)* 147–169
24. Fortune, S.: A sweepline algorithm for voronoi diagrams. *Algorithmica* **2** (1987) 153–174
25. S. Raupp Musse, D.: A behavioral model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics* **7** (2001) 152–164

Referencias

- [1] MM Samson, A Crowe, PL De Vreede, JAG Dessens, SA Duursma, and HJJ Verhaar. Differences in gait parameters at a preferred walking speed in healthy subjects due to age, height and body weight. *Aging Clinical and Experimental Research*, 13(1):16–21, 2001. [iii](#), [20](#), [21](#)
- [2] Francisco Resendiz. Funeral de hugo chavez. In *El Universal*, 2013. [v](#), [3](#)
- [3] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, August 1987. [v](#), [10](#), [12](#)
- [4] Craig W. Reynolds. Steering behaviors for autonomous characters. In *1999 Game Developers Conference*, pages 763–782, 1999. [v](#), [11](#), [12](#)
- [5] Zvi Shiller Paolo Fiorini. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, pages 760–772, July 1998. [v](#), [13](#), [14](#)
- [6] Stephen Chenney. Flow tiles. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA*, pages 233–242, 2004. [v](#), [2](#), [14](#)
- [7] Julien Pettré, Pablo de Heras Ciechowski, Jonathan Maïm, Barbara Yersin, Jean-Paul Laumond, and Daniel Thalmann. Real-time navigating crowds: scalable simulation and rendering. *Computer Animation and Virtual Worlds*, 17(3-4):445–455, 2006. [v](#), [15](#)
- [8] Jur Van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1928–1935. IEEE, 2008. [v](#), [13](#), [15](#), [16](#)

REFERENCIAS

- [9] Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics research*, pages 3–19. Springer, 2011. [v](#), [16](#)
- [10] Rebollar Fernando, Ramos Marco, Muñoz Vianney, and Ramos Félix. Generación de multitudes virtuales heterogéneas basadas en patrones de agrupación de comportamiento. *Research in Computer Science*, 124:95–107, 2016. [v](#), [v](#), [v](#), [vi](#), [27](#), [31](#), [32](#), [33](#)
- [11] Yoselie Alvarado, Jacqueline Fernández, and Roberto A Guerrero. Vida artificial y personajes virtuales. In *XVI Workshop de Investigadores en Ciencias de la Computación*, 2014. [1](#), [7](#)
- [12] Oriam De Gyves, Leonel Toledo, and Isaac Rudomin. Comportamientos en simulación de multitudes: revisión del estado del arte. *Research in Computer Science*, pages 319–334, 2013. [1](#), [3](#), [11](#), [12](#)
- [13] Oriam De Gyves, Leonel Toledo, Iván Rivalcoba, and Isaac Rudomin. Simulación de grandes multitudes con dinámica de grupos. *Research in Computer Science*, 2015. [1](#), [17](#)
- [14] Erik Millan, Benjamin Hernández, and Isaac Rudomin. Large crowds of autonomous animated characters using fragment shaders and level of detail. *ShaderX5: Advanced Rendering Techniques*, pages 501–510, 2007. [2](#), [11](#), [13](#)
- [15] Shiizuka H. Takahashi, T. Behavior simulation by network model. *Memoirs of Kougakuin University*, pages 213–220, 1992. [2](#)
- [16] Vicsek T Helbing D., Farkas I. Simulating dynamical features of escape panic. *Nature* 407, 2000. [2](#)
- [17] Daniel Thalmann, Helena Grillon, Jonathan Maim, and Barbara Yersin. Challenges in crowd simulation. In *2009 International Conference on Cyber-Worlds*, pages 1–12. IEEE, 2009. [2](#)
- [18] D.Thalmann S. Raupp Musse. A behavioral model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):152–164, 2001. [2](#), [32](#)
- [19] Mubbasir Kapadia, Shawn Singh, Glenn Reinman, and Petros Faloutsos. A behavior-authoring framework for multiactor simulations. *Computer Graphics and Applications, IEEE*, 31(6):45–55, 2011. [2](#), [11](#)

-
- [20] Nuria Pelechano, Jan M Allbeck, and Norman I Badler. Controlling individual agents in high-density crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 99–108. Eurographics Association, 2007. 2, 11
- [21] Daniel Thalmann. Populating virtual environments with crowds. In *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*. ACM, 2006. 2, 15
- [22] Sergio Ruiz and Benjamin Hernández. Procesos de decisión de markov y microescenarios para navegación y evasión de colisiones para multitudes. *ResearchGate*, 2014. 2, 12
- [23] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 2
- [24] Lebon G. *Psychologie des foules*. Alcan, Paris, 1895. 2
- [25] Saskia Groenewegen. Improving crowd behaviour for games and virtual worlds. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, pages 256–258. ACM, 2010. 3
- [26] Michael Wißner, Felix Kistler, and Elisabeth André. Level of detail ai for virtual characters in games and simulation. In *Motion in Games*, pages 206–217. Springer, 2010. 3
- [27] Oliver Szymanczyk, Patrick Dickinson, Tom Duckett, et al. From individual characters to large crowds: augmenting the believability of open-world games through exploring social emotion in pedestrian groups. *ARA Digital Media Private Limited*, 2011. 3
- [28] Yanbin Wang, Rohit Dubey, Nadia Magnenat-Thalmann, and Daniel Thalmann. An immersive multi-agent system for interactive applications. *The Visual Computer*, 29(5):323–332, 2013. 3
- [29] Ute Ritterfeld, Michael Cody, and Peter Vorderer. *Serious games: Mechanisms and effects*. Routledge, 2009. 3
- [30] José Jesús Martínez Páez et al. Agentes autónomos inteligentes. *Revista de la Facultad de Medicina; Vol. 48, núm. 3;*, pages 168–169, 2000. 3
- [31] Dashun Wang, Dino Pedreschi, Chaoming Song, Fosca Giannotti, and Albert-Laszlo Barabasi. Human mobility, social ties, and link prediction. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1100–1108. ACM, 2011. 3

REFERENCIAS

- [32] Zhi Zhong, Ning Ding, Xinyu Wu, and Yangsheng Xu. Crowd surveillance using markov random fields. In *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, pages 1822–1828. IEEE, 2008. 3
- [33] Ramin Mehran, Akira Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 935–942. IEEE, 2009. 3
- [34] Tian Cao, Xinyu Wu, Jinnian Guo, Shiqi Yu, and Yangsheng Xu. Abnormal crowd motion analysis. *ROBIO*, 9:1709–1714, 2009. 3
- [35] Muhammad Husni and Nanna Suryana. Crowd event detection in computer vision. In *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, volume 1, pages V1–444. IEEE, 2010. 3
- [36] Andreas Zweng and Martin Kampel. Unexpected human behavior recognition in image sequences using multiple features. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 368–371. IEEE, 2010. 3
- [37] Guogang Xiong, Xinyu Wu, Yen-Lun Chen, and Yongsheng Ou. Abnormal crowd behavior detection based on the energy model. In *Information and Automation (ICIA), 2011 IEEE International Conference on*, pages 495–500. IEEE, 2011. 3
- [38] Nan Li and Zhimin Zhang. Abnormal crowd behavior detection using topological methods. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2011 12th ACIS International Conference on*, pages 13–18. IEEE, 2011. 3
- [39] Jesse Schell. *The Art of Game Design: A book of lenses*. CRC Press, 2014. 5
- [40] David R Michael and Sandra L Chen. *Serious games: Games that educate, train, and inform*. Muska & Lipman/Premier-Trade, 2005. 5
- [41] Beatriz E Marcano Lárez. Juegos serios y entrenamiento en la sociedad digital. 2008. 5
- [42] Ben Sawyer and Peter Smith. Serious games taxonomy. In *Slides from the Serious Games Summit at the Game Developers Conference*, 2008. 5
- [43] Lennart Nacke. Affective ludology: Scientific measurement of user experience in interactive entertainment. 2009. 6

-
- [44] Victor Fernández Cervantes. *Interpretación de Estructura y Movimiento Corporal Usando Teoría de Patrones Difusa*. PhD thesis, Centro de Investigación y Estudios Avanzados del I. P. N., Unidad Guadalajara, Junio 2014. [6](#)
- [45] Andrew J Stapleton. Serious games: Serious opportunities. In *Australian Game Developers? Conference, Academic Summit, Melbourne*, 2004. [6](#)
- [46] Mohammad A Fraiwan, Natheer Khasawneh, Ahmed Malkawi, Muhammed Al-Jarrah, R Alsa'di, and S Al-Momani. Therapy central: On the development of computer games for physiotherapy. In *Innovations in Information Technology (IIT), 2013 9th International Conference on*, pages 24–29. IEEE, 2013. [6](#)
- [47] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009. [6](#)
- [48] Mark De Berg, Marc Van Kreveld, Mark Overmars, and Otfried Cheong Schwarzkopf. Computational geometry. In *Computational geometry*, pages 147–169. Springer, 2000. [8](#)
- [49] Steven Fortune. A sweepline algorithm for voronoi diagrams. *Algorithmica*, 2(1-4):153–174, 1987. [10](#), [26](#)
- [50] Libo Sun and Wenhui Qin. Simulation of crowd behaviors based on event reaction. In *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on*, volume 2, pages 163–167. IEEE, 2011. [11](#)
- [51] Sunil Hadap, Dave Eberle, Pascal Volino, Ming C Lin, Stephane Redon, and Christer Ericson. Collision detection and proximity queries. In *ACM SIGGRAPH 2004 Course Notes*, page 15. ACM, 2004. [12](#)
- [52] Stephen J Guy, Jatin Chhugani, Sean Curtis, Pradeep Dubey, Ming Lin, and Dinesh Manocha. Pedestrians: a least-effort approach to crowd simulation. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 119–128. Eurographics Association, 2010. [13](#)
- [53] Byn Choi, Rakesh Komuravelli, Victor Lu, Hyojin Sung, Robert L Bocchino, Sarita V Adve, and John C Hart. Parallel sah kd tree construction. In *Proceedings of the Conference on High Performance Graphics*, pages 77–86. Eurographics Association, 2010. [13](#)

REFERENCIAS

- [54] Artur L dos Santos, Joao Marcelo XN Teixeira, Thiago SMC De Farias, Veronica Teichrieb, and Judith Kelner. Kd-tree traversal implementations for ray tracing on massive multiprocessors: a comparative study. In *Computer Architecture and High Performance Computing, 2009. SBAC-PAD'09. 21st International Symposium on*, pages 41–48. IEEE, 2009. [13](#)
- [55] Avi Bleiweiss. Multi agent navigation on the gpu. In *GDC09 Game Developers Conference*, volume 2009, 2009. [13](#)
- [56] Thomas Jund, Pierre Kraemer, and David Cazier. A unified structure for crowd simulation. *Computer Animation and Virtual Worlds*, 23(3-4):311–320, 2012. [13](#)
- [57] Erick Baptista Passos, Mark Joselli, Marcelo Zamith, Esteban Walter Gonzalez Clua, Anselmo Montenegro, Aura Conci, and Bruno Feijo. A bidimensional data structure and spatial optimization for supermassive crowd simulation on gpu. *Computers in Entertainment (CIE)*, 7(4):60, 2009. [13](#)
- [58] Kenneth E Hoff III, Andrew Zaferakis, Ming Lin, and Dinesh Manocha. Fast and simple 2d geometric proximity queries using graphics hardware. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 145–148. ACM, 2001. [13](#)
- [59] Adrien Treuille, Seth Cooper, and Zoran Popović. Continuum crowds. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1160–1168. ACM, 2006. [15](#)
- [60] Leonardo G Fischer, Renato Silveira, and Luciana Nedel. Gpu accelerated path-planning for multi-agents in virtual environments. In *Games and Digital Entertainment (SBGAMES), 2009 VIII Brazilian Symposium on*, pages 101–110. IEEE, 2009. [15](#)
- [61] Fábio Dapper, Edson Prestes, and Luciana P Nedel. Generating steering behaviors for virtual humanoids using bvp control. In *Proc. of CGI*, volume 1, pages 105–114, 2007. [15](#)
- [62] Hubert Klüpfel. The simulation of crowds at very large events. In *Traffic and Granular Flow'05*, pages 341–346. Springer, 2007. [15](#)
- [63] Shuai Zhang, Manchun Li, Feixue Li, Aili Liu, and Dong Cai. A simulation model of pedestrian flow based on geographical cellular automata. In *Geoinformatics, 2011 19th International Conference on*, pages 1–5. IEEE, 2011. [16](#)

- [64] Daniel Thalmann Yanbin Wang, Rohit Dubey. An immersive multi-agent system for interactive applications. *The Visual Computer*, pages 323–332, June 2013. [16](#)
- [65] Oriam De Gyves, Leonel Toledo, and Isaac Rudomín. Proximity queries for crowd simulation using truncated voronoi diagrams. In *Proceedings of Motion on Games*, pages 87–92. ACM, 2013. [17](#)
- [66] Manuel Domínguez-Rodrigo. *El origen del comportamiento humano*. Librería Tipo, 1994. [20](#)
- [67] Joao Claudio Todorov. Behavioral analysis of non-experimental data associated with cultural practices. *Behavior and Social Issues*, 18:10, 2009. [20](#)
- [68] Organización Mundial de la Salud. La salud de los jóvenes: un desafío para la sociedad. *Organización Mundial de la Salud, Ginebra 1986*, (731), 2000. [27](#)