

UNIVERSIDADE FEDERAL DE PERNAMBUCO  
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO  
CENTRO DE INFORMÁTICA

---

# **Simulando Multidões Virtuais**

---

TRABALHO DE GRADUAÇÃO

Junho de 2009

# Simulando Multidões Virtuais

por

Felipe Maia

Monografia apresentada ao Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do Grau de Bacharel em Ciência da Computação.

**Orientador:** Silvio de Barros Melo

## **Agradecimentos**

Dedico este trabalho e agradeço à minha amada Michelle, pelo zelo, compreensão e incentivo durante estes quatro anos e meio de estudo e trabalho. Agradeço a minha família pela educação e incentivo que me foi proporcionado durante toda a minha vida, sem as quais eu não chegaria onde cheguei. Agradeço aos colegas e amigos pela colaboração e incentivo mútuo. Obrigado a Silvio Melo, pela paciência, dedicação e clareza que desprendeu desde o início do curso. Obrigado a Marcelo Walter, pela fomentação do espírito de pesquisa durante o andamento do curso e pelo incentivo e. Agradeço a meus amigos e colegas de trabalho pelo incentivo e ajuda, em especial a Tulio Caraciolo, pela ajuda em todo o processo do trabalho, e a Dirac Vasconcelos, pelos modelos fornecidos que infelizmente não foram utilizados.

Muito obrigado!

## Resumo

A simulação computacional de multidões é uma área da computação que cresce constantemente, seja em produções cinematográficas, jogos digitais, ou simulações para estudos específicos como, por exemplo, planejamento de construções. Uma completa simulação virtual de multidões inclui muitos problemas. Defrontando-se com esta complexidade, este trabalho tem como proposta realizar um estudo sobre todos os aspectos da simulação de multidões virtuais, com foco principal no aspecto comportamental da simulação.

Neste sentido, foi escolhido para implementação um modelo em tempo real baseado na dinâmica do contínuo, proposto por Treuille et al. [39]. O modelo utiliza um campo de potencial dinâmico para integrar o planejamento de trajetória global com obstáculos móveis como outras pessoas, eficientemente resolvendo a locomoção de grandes multidões sem a necessidade explícita de tratamento de colisão. Nossas simulações rodam em tempo real, apresentam fluxo de movimentação suave sobre uma variedade de situações, e naturalmente exibem fenômenos emergentes que são observados em multidões reais.

**Palavras chave:** Multidões, Simulação de seres humanos, Comportamento humano, Pedestres, Planejamento de trajeto, Ambientes virtuais.

# Sumário

1	Introdução .....	7
2	Multidões Virtuais .....	8
2.1	Requisitos e Restrições para a Modelagem de Multidões .....	9
2.2	Áreas que englobam a simulação de multidão .....	9
2.3	Aplicações.....	10
2.3.1	Arquitetura e segurança.....	10
2.3.2	Sistemas de treinamento de gerenciamento de multidões.....	11
2.3.3	Modelos sociológicos de multidões .....	12
2.3.4	Comportamento em grupo em robótica e inteligência artificial .....	13
2.3.5	Multidões em produções cinematográficas.....	13
2.3.6	Multidões em Jogos .....	15
2.4	Multidões em mundos virtuais .....	16
2.5	Comportamento e animação de grupos e multidões .....	17
2.6	Renderização de multidões.....	19
2.7	Modelagem do ambiente para multidões .....	21
3	Implementação .....	23
3.1	Abordagem do comportamento .....	24
3.1.1	O Modelo.....	24
3.1.2	Implementando o modelo.....	27
3.1.2.1	Conversão da densidade .....	28
3.1.2.2	Custo unitário .....	28
3.1.2.3	Construindo o campo de potencial dinâmico .....	29
3.1.2.4	Movimentando a multidão.....	30
3.1.2.5	Garantindo a distância entre as pessoas.....	30
3.1.3	Estrutura de classes.....	30
3.2	Abordagem da renderização .....	31
3.3	Abordagem do ambiente .....	31
4	Resultados Obtidos .....	32
5	Considerações finais.....	35
6	Referencias.....	36

## Lista de Figuras

Figura 2.1: Densidade da multidão e visualização da simulação em Simulex [44]. .....	11
Figura 2.2: Cena do filme <i>The Lord of the Ring: the Two Towers</i> , criada com a ajuda do programa de simulação de multidões Massive.....	14
Figura 2.3: Screenshot do programa Massive, utilizado em produções cinematográficas. ....	15
Figura 2.4: Peixes artificiais apresentados por Tu e Terzopoulos [47].....	17
Figura 2.5: Geração de variedade utilizando modelos distintos e modulação de textura. ....	19
Figura 3.1: Resultado do modelo continuum crowds apresentado por Treuille et al [39]. ....	23
Figura 3.2: Estrutura de grade discretizada .....	27
Figura 3.3: Estrutura do cálculo da densidade .....	28
Figura 3.4: Visão geral do algoritmo apresentada no artigo original.....	30
Figura 4.1: Dois grupos atravessam um corredor, naturalmente formando raias: resultado apresentado no artigo continuum crowds.....	32
Figura 4.2: Nosso resultado de um cenário similar, apresentando corretamente o mesmo fenômeno de formação de raias. ....	32
Figura 4.3: Quatro grupos se cruzam, formando um padrão de vórtex: resultado apresentado no artigo continuum crowds. ....	33
Figura 4.4: Nosso resultado de um cenário similar, apresentando corretamente o mesmo fenômeno de formação de padrões de vórtex. ....	33
Figura 4.5: Exemplo de cenário criado por nós: cada objetivo é representado por uma cor. Os obstáculos estão representados em preto. ....	34

# 1 Introdução

À medida que nós atravessamos nosso cotidiano, nossas vidas se intersectam com outras pessoas. Nós vemos pessoas saindo para o trabalho, esperando por trens, encontrando-se com amigos, trabalhando em seus empregos, e realizando diversas outras atividades. As pessoas criam um rico tapete de atividades em seus cotidianos, uma *textura humana*. Nós podemos não estar sempre conscientes desta textura, porém nós definitivamente a notaríamos se ela estivesse ausente.

Existem muitas aplicações computacionais de animação e simulação nas quais é necessária a modelagem de multidões virtuais. Algumas dessas aplicações incluem planejamento de ambientes, educação, entretenimento, treinamento, e análise de evacuação. Algumas outras aplicações incluem simulações de cenários onde uma grande quantidade de pessoas se encontra, se movimenta, e se dispersa, tais como em centros de transporte, eventos esportivos, e concertos. Algumas vezes essas populações podem estar mutuamente engajadas em uma atividade comum, como por exemplo a evacuação de uma construção ou área; outras vezes elas podem estar focadas em suas agendas pessoais de trabalho, entretenimento, descanso, viagem ou observação.

O comportamento humano individual em sua forma mais elementar é difícil de ser modelado. Caso consideremos vários indivíduos e suas relações entre si, a complexidade do sistema pode substancialmente aumentar. A área de simulação de multidões é bastante complexa e compreende problemas como planejamento de trajeto, tratamento de colisão, inter-relacionamento dos indivíduos com formação de grupos distintos, diferenciação visual entre cada indivíduo, e muitos outros.

Este trabalho tem como objetivo a investigação da simulação de multidões virtuais. Nós apresentaremos todos os aspectos relativos à simulação de multidões, incluindo a animação comportamental e renderização. Então apresentaremos a implementação de um simulador de multidões com foco no modelo comportamental, um modelo macroscópico que integra em tempo real a navegação global com pessoas e obstáculos se movendo através de um campo de potencial dinâmico.

No Capítulo 2 nós inicialmente apresentamos uma visão geral da simulação de multidões, discutimos seus requisitos e restrições, enumeramos as áreas envolvidas em sua simulação, apresentamos as aplicações existentes separadas por área, e finalmente resumimos o estado da arte nas áreas de comportamento, renderização, e modelagem de ambiente. No Capítulo 3, nós apresentamos o modelo que escolhemos para nossa implementação, inicialmente nós resumimos a teoria na qual ele se baseia, posteriormente explicamos detalhadamente sua implementação, finalmente explicamos os outros detalhes de implementação, incluindo renderização e modelagem do ambiente. No Capítulo 4, nós apresentamos os resultados obtidos por nossa implementação. Finalmente, no Capítulo 5, encontram-se as conclusões do trabalho, e são discutidas as dificuldades apresentadas durante o desenvolvimento do trabalho.

## 2 Multidões Virtuais

Apesar de o comportamento coletivo haver sido estudado desde o final do século dezenove [23], tentativas de simulá-lo através de modelos computacionais são recentes, com a maioria dos trabalhos sendo feita no meio e no final dos anos noventa.

A maioria das abordagens para a modelagem de multidões é específica à aplicação alvo, focando em aspectos distintos do comportamento coletivo e utilizando diferentes técnicas de modelagem. As técnicas utilizadas abrangem as que não distinguem os indivíduos entre si como modelos de redes ou fluidos, e aquelas que representam cada indivíduo sendo controlado por alguma regra, simples ou complexa, como leis físicas, equações do caos, modelos comportamentais, ou simulações sociológicas.

É possível distinguir duas amplas áreas na simulação de multidões. A primeira tem como foco o realismo dos aspectos comportamentais, utilizando uma visualização simples, comumente 2D, como simuladores de evacuação, modelos sociológicos de multidões ou modelos dinâmicos de multidões. Nesta área, o comportamento simulado é geralmente realizado com um estreito e controlado objetivo (por exemplo, pessoas simplesmente correndo para a saída ou formando estruturas mais complexas de interação, como a formação de anéis) com esforços focados na validação da correspondência destes eventos em situações do mundo real [44]. Idealmente, os resultados destas simulações seriam consistentes com observações de multidões reais. A visualização é utilizada para ajudar a entender os resultados da simulação, mas não é crucial. Na maioria dos casos, uma representação esquemática, com os membros das multidões representados por pontos coloridos ou figuras estáticas é suficiente, e algumas vezes até preferíveis, permitindo o destaque de informações mais relevantes.

Na segunda área, o principal objetivo é a visualização de alta qualidade da multidão (por exemplo, em produções cinematográficas ou em jogos eletrônicos), mas geralmente o realismo fiel do comportamento do modelo não é uma prioridade. O importante é um resultado visual convincente, que é parcialmente obtido por modelos de comportamento e parcialmente pela intervenção humana no processo de produção. Uma multidão virtual necessita igualmente ser animada de uma maneira acreditável e ter uma boa aparência aos olhos humanos, tendo como ênfase da pesquisa métodos para a animação e renderização da multidão. Indivíduos da multidão são visualizados como figuras tridimensionais com textura, iluminação e animação, de forma a se encaixar ao ambiente ao qual está inserido [9]. Nessa área, modelos de comportamento não tentam necessariamente se igualar ao mundo real, seu propósito é mais ser capaz de aliviar o trabalho dos animadores e possibilitar resposta às entradas em aplicações interativas.

Todavia, a tendência atual é uma convergência de ambas as áreas, onde sistemas orientados à visualização estão procurando incorporar modelos de comportamento mais complexos, capazes de facilitar a criação de animações convincentes; e os sistemas com foco no comportamento estão tentando obter uma melhor visualização, especialmente no domínio dos simuladores de evacuação. As aplicações que mais demandam de ambas as áreas são os sistemas de treinamento, onde a visualização é tão importante como o realismo comportamental para que o treinamento seja efetivo.



## 2.1 Requisitos e Restrições para a Modelagem de Multidões

A simulação de multidões em tempo real possui dificuldades distintas das apresentadas em sistemas que envolvem uma pequena quantidade de indivíduos (como é o caso da maioria dos jogos contemporâneos) ou das aplicações que não são em tempo real (como multidões em filmes ou simulações complexas de evacuação).

A principal diferença em relação a simulações de poucos indivíduos é a necessidade de gerenciamento eficiente das variedades em todos os níveis, seja visualização, controle de movimentação, animação ou comportamento. Como na vida real, cada indivíduo deve parecer diferente, andar diferente, reagir diferente, se comportar diferente, e assim por diante. Mesmo que fosse possível a simulação perfeita de um humano virtual, criar uma simulação com múltiplos destes humanos seria uma tarefa difícil. Métodos permitindo o controle de múltiplos agentes ao mesmo tempo são necessários, no entanto, tais métodos deveriam preservar a habilidade de controlá-los individualmente.

Em comparação com simulações que não possuam a necessidade de executar em tempo real, a principal dificuldade é uma maior demanda de poder computacional, seja poder de processamento geral, desempenho gráfico, ou espaço na memória. Um dos maiores fatores limitantes para a simulação de multidões em tempo real é a renderização. Métodos rápidos e escaláveis para computar o comportamento, capazes de levar em consideração entradas previamente desconhecidas, e para renderizar multidões grandes e variadas são necessários. Enquanto simulações *off-line* possuem a vantagem de conhecer a simulação por completo (e portanto, por exemplo, podem rodar iterativamente sobre várias possíveis opções, selecionando a melhor solução), simulações em tempo real têm de reagir à situação a medida que ela é apresentada.

## 2.2 Áreas que Englobam a Simulação de Multidões

Para a completa simulação de uma multidão em um ambiente virtual, é necessário resolver diversos problemas. As áreas de relevância para a simulação de multidão incluem:

- Geração de indivíduos virtuais: Como gerar uma multidão heterogênea? Como criar uma população com uma distribuição de características desejadas?
- Animação da multidão: Como as entidades virtuais devem se movimentar ao redor e evitar colisões com o ambiente estático e objetos dinâmicos? Como um grupo pode se locomover de maneira coordenada?
- Formação do comportamento da multidão: Como uma multidão virtual deve responder a mudanças em seu ambiente? Como os agentes devem responder ao comportamento de outros agentes? Qual é uma maneira eficiente de modelar a percepção para muitos agentes?
- Interação com multidões virtuais: Como e qual informação deve ser dada por pessoas reais para controlar as entidades virtuais? Qual é a mais eficiente maneira para dirigir figurantes virtuais?
- Renderização de multidões virtuais: Como mostrar muitos indivíduos animados de forma rápida? Como mostrar uma grande variedade de aparências?

- Integração das multidões com os ambientes virtuais: Quais aspectos do ambiente precisam ser modelados? Qual é a representação mais adequada para representar os objetos que compõem o ambiente de maneira a minimizar a computação do comportamento?

Muitos desses aspectos são interligados. Por exemplo, eficiência na renderização restringe a variedade de comportamentos e aparências; geração de comportamento em alto-nível controla a movimentação em baixo nível, porém o comportamento deve também responder a colisões durante a locomoção de forma apropriada; o modelo do comportamento afeta as possibilidades de interação com uma pessoa real; a representação do ambiente afeta os possíveis comportamentos.

## 2.3 Aplicações

Nas últimas duas décadas, pesquisadores de uma ampla variedade de áreas como arquitetura, computação gráfica, física, robótica, ciência da segurança, sistemas de treinamento e sociologia têm criado simulações envolvendo uma coleção de indivíduos. Nesta seção, nós enumeramos algumas das áreas, e alguns de seus respectivos sistemas, que realizam a simulação de multidões. Para maiores detalhes dos sistemas de simulação existentes, nós referimos o leitor aos *surveys* realizados por Still [38], Teknomo [41], e Kilogowski e Peacock [21], e aos recentes livros da área de multidões virtuais [54, 55].

### 2.3.1 Arquitetura e segurança

Uma das maiores áreas onde o comportamento de multidões foi modelado é a área de segurança e arquitetura, tendo como aplicação dominante a de simuladores de evacuação de multidões. Tais sistemas modelam a movimentação de uma grande quantidade de pessoas em locais geralmente fechados e bem definidos, como ambientes internos de construções, metrô, barcos, ou aviões. A sua meta é ajudar os designers e arquitetos a entender a relação entre a organização do espaço e o comportamento humano [29].

Os simuladores de evacuação são mais comumente utilizados para modelar o comportamento de uma multidão em caso de uma evacuação forçada de um ambiente confinado devido a alguma ameaça como fogo ou fumaça. Nessa situação, uma quantidade de pessoas tem de evacuar a área determinada, geralmente através de um número relativamente pequeno de saídas. As simulações tentam responder perguntas como: A área pode ser evacuada em um tempo pré-definido? Onde os gargalos do fluxo de pessoas acontecem? Quais são as prováveis áreas onde uma súbita aglomeração de pessoas possa causar uma pressão de esmagamento inaceitável? A abordagem mais comum de modelagem nessa área é o uso de autômatos celulares para representar não só os indivíduos, mas também o ambiente.

*Exodus* [13] é um sistema capaz de simular a evacuação de uma grande quantidade de indivíduos através de grandes construções de vários pavimentos. Utilizando modelos de dinâmica de fluidos, juntamente com técnicas discretizadas de simulação de realidade virtual, o programa rastreia a trajetória dos indivíduos enquanto eles tentam evacuar a área designada ou são superados pelas ameaças (por exemplo, fogo e fumaça). As saídas de *Exodus* incluem o tempo de evacuação total, tempos individuais de espera e evacuação, e as trajetórias de evacuação efetuadas pelos indivíduos [13].

A empresa *London Underground Limited* desenvolveu originalmente um sistema chamado *Pedroute*, utilizado extensivamente para simular parâmetros de multidões em redes subterrâneas ao redor do mundo [6]. É um modelo de maximização de entropia espacial que tem sido utilizado para o design de estações, incluindo o *Olympic Railway Station* em *Sidney* (criado com o objetivo de manusear 50,000 passageiros em uma hora). O modelo pode simular a movimentação de trens e passageiros através de uma estação ou construção. O desempenho da estação é avaliado através dos níveis de serviços, densidade dos passageiros e atrasos, fornecendo estatísticas do tempo de viagem dos passageiros, congestionamentos e nível de serviço de cada segmento da estação.

*Simulex* [44] é um modelo computacional que simula a evacuação de pessoas através de grandes construções complexas geometricamente, definidas por plantas baixas e as escadarias que as conectam. Cada indivíduo possui diversos atributos como posição, tamanho de corpo, ângulo de orientação e velocidade. Vários algoritmos como função de distância, busca de caminho, ultrapassagem, desvio de rota e ajuste de velocidade de acordo com a proximidade de outros indivíduos são utilizados para simular a egressão, onde os ocupantes caminham em direção e através das saídas.

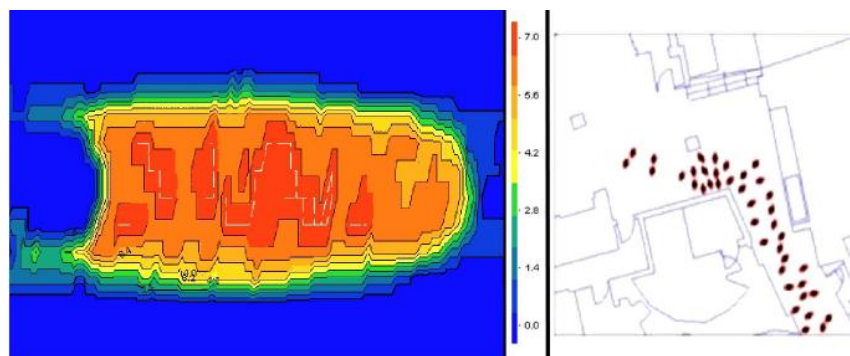


Figura 2.1: Densidade da multidão e visualização da simulação em *Simulex* [44].

G. Still desenvolveu uma coleção de programas chamada *Legion* para a simulação e análise da dinâmica de multidões em evacuações de ambientes complexos e restritos como estádios [38]. As dinâmicas de locomoção da multidão são modeladas através de autômatos celulares móveis. O modelo emprega quatro parâmetros (local objetivo, velocidade, distancia aos outros indivíduos, e tempo de reação) e uma regra de decisão para simular os indivíduos. O modelo trabalha com o espaço 2D contínuo, produzindo caminhos mais realistas para os pedestres do que os modelos que utilizam discretizações do ambiente.

### 2.3.2 Sistemas de treinamento de gerenciamento de multidões

A modelagem de multidões também é essencial aos sistemas de simulação usados para o treinamento de policiais e militares, ensinando-os como lidar com uma aglomeração em massa de pessoas.

*CACTUS* [52] é um sistema desenvolvido para auxiliar o planejamento e treinamento durante incidentes de ordem pública como grandes demonstrações e manifestações. O funcionamento do programa é baseado em um modelo de mundo ao qual são inseridos grupos de multidões e unidades policiais no mapa digitalizado, que possuem regras probabilísticas para modelar seus comportamentos e interações. O modelo de simulação representa pequenos grupos de

peessoas como objetos discretos. As descrições comportamentais estão representadas na forma de um grafo direcionado no qual os nós representam estados de comportamento (aos quais correspondem ações e emoções exibidas) e as arestas representam mudanças plausíveis entre esses estados. As transições entre estados dependem das condições ambientais e dos pesos das probabilidades. A simulação roda como um exercício de tomada de decisão a qual pode incluir planejamento logístico pré-determinado, gerenciamento de incidentes, e avaliação de relatório posterior.

*Small Unit Leader Non-Lethal Training System* [51] é um simulador para o treinamento do U.S Marine Corps (marinha dos estados unidos) na tomada de decisão com respeito ao uso de munições não letais em operações de manutenção de paz e controle de multidões. Os *trainees* aprendem as regras de abordagem, os procedimentos para lidar com multidões, e a habilidade de tomar decisões sobre o nível apropriado de força necessário para controlar, conter e dispersar multidões. Multidões andam pelo ambiente urbano simulado por caminhos pré-definidos por um instrutor, e respondem tanto às ações dos *trainees* como às ações de outras multidões. Cada multidão é caracterizada por um perfil, contendo uma série de atributos como fanatismo, nervosismo, experiência anterior com armas não letais, ou atitude voltada aos *marines*. Durante o exercício, o modelo computado opera em tempo real e responde às ações e inações do *trainee* com o comportamento apropriado, como: celebração, demonstração, revolta, e dispersão de acordo com uma série de relações booleanas definidas por especialistas.

### **2.3.3 Modelos sociológicos de multidões**

Apesar de ser um campo essencialmente preocupado em estudar o comportamento coletivo, somente uma pequena quantidade de trabalhos foi realizada na área de sociologia.

McPhail et al. [24] estudou ações individuais e coletivas em aglomerações temporárias de pessoas. O modelo da multidão é baseado na teoria do controle de percepção, onde cada indivíduo tenta controlar sua experiência de forma a manter um relacionamento com outros: nesse caso, é um relacionamento espacial com outros integrantes do grupo. O programa de simulação GATHERING mostra graficamente a movimentação e o surgimento de estruturas dentro da multidão. O mesmo sistema de simulação foi utilizado posteriormente por Schwein-gruber [35] para estudar os efeitos de sinais de referências, comuns na coordenação do comportamento coletivo, e por Tucker et al. [46] para estudar a formação de arcos e anéis em aglomerações temporárias.

Jager et al. [17] modelou o agrupamento e enfrentamento em multidões de dois bandos. Uma multidão é modelada por uma simulação multi-agente utilizando autômatos celulares com regras definindo conflito de aproximação/afastamento. A simulação consiste de dois grupos de agentes de três tipos distintos: *hardcore*, *hangers-on*, and *bystanders*, tendo como diferença entre eles a frequência com a qual eles sondam sua vizinhança. O objetivo da simulação era estudar os efeitos do tamanho do grupo, simetria do tamanho, e a composição do grupo em agrupamentos e confrontos.

### **2.3.4 Comportamento em grupo em robótica e inteligência artificial**

Pesquisadores trabalhando na área de inteligência artificial estão interessados em explorar como o comportamento coletivo emerge de regras de comportamentos locais [14]. Modelos de software e grupos de robôs foram concebidos com o objetivo de realizar diversas experiências para entender como o comportamento coletivo complexo pode surgir de sistemas guiados por regras simples. A maior fonte de inspiração é a natureza, onde, por exemplo, insetos sociais eficientemente resolvem problemas como encontrar comida, construir ninhos, ou dividir o trabalho entre si, através de simples interações entre indivíduos sem uma entidade controladora global. Um mecanismo importante para o controle dividido do comportamento é a utilização de feromônios, interações indiretas entre indivíduos são realizadas através da modificação do ambiente [3].

Dorigo introduziu sistemas de formigas inspirado pelo comportamento de colônias de formigas reais [10]. Algoritmos com abordagem de formigas tem sido utilizados com sucesso para a solução de uma variedade de problemas de otimização discreta, incluindo o problema do caixeiro viajante, ordenamento sequencial, coloração de grafos e roteamento de redes [2]. Além de insetos, grupos de organismos mais complexos como bando de aves, rebanhos de animais e escolas de peixes têm sido estudados com o objetivo de conhecer seus princípios e organização. Couzin et al. apresentou um modelo de como animais que exploram o ambiente em busca de alimento ou viajam em grupos podem tomar decisões mesmo com uma pequena quantidade de indivíduos informados [7].

Princípios de sistemas biológicos também foram utilizados para controlar o comportamento de grupos autônomos compostos de robôs. Diversos são os trabalhos em que o comportamento de robôs é inspirado por comportamentos observados na natureza, como colméias, formigas, e até pedestres humanos. Trabalhos que utilizam robôs para influenciar o comportamento de animais reais também existem; Vaughan et al. [50] desenvolveu um robô móvel capaz de agrupar um bando de patos reais e manobrá-los com segurança para uma posição final designada.

### **2.3.5 Multidões em produções cinematográficas**

Um dos domínios com o maior crescimento da simulação de multidões é o de efeitos especiais. Enquanto há cerca de uma década atrás não existiam multidões digitais, hoje quase todas as superproduções cinematográficas apresentam algum tipo de multidão virtual, com a televisão e propaganda começando a acompanhar esta tendência. Em comparação com multidões de figurantes reais, multidões virtuais permitem a redução significativa dos custos da produção de cenas com muitos indivíduos e permitem uma maior liberdade criativa devido à sua flexibilidade. Técnicas distintas, como a replicação de filmagens do mundo real, sistemas de partícula ou animação comportamental foram utilizadas para adicionar multidões de figurantes virtuais em cenas de diversos tipos de filme, como dramas históricos, ficções científicas ou fantásticas e desenhos animados.

O fator principal determinante para a escolha das técnicas a serem utilizadas é a qualidade visual final e os custos de produção permitidos ao projeto. É comum o uso de técnicas distintas em uma mesma cena de maneira a obter os melhores resultados visuais. Por exemplo,

personagens no plano frontal são atores reais, com personagens virtuais 3D assumindo papéis secundários no plano de fundo.



**Figura 2.2:** Cena do filme *The Lord of the Ring: the Two Towers*, criada com a ajuda do programa de simulação de multidões Massive.

Apesar de uma considerável quantidade de trabalho ter sido feita na simulação de multidões na área cinematográfica, somente pouca informação é disponibilizada, principalmente a respeito de detalhes mais técnicos. A maioria do conhecimento disponibilizado é proveniente de fontes distintas, tais como documentários “*making-of*”, entrevistas com integrantes da equipe de efeitos especiais, ou resenhas jornalísticas da indústria. Para produções de grande orçamento, a abordagem mais comum é o desenvolvimento próprio de ferramentas customizadas, as quais são utilizadas para a produção de um determinado filme. Como a qualidade da animação é suprema, grandes bibliotecas de *motion-clips* são geralmente utilizadas, obtidas principalmente da captura de movimento de atores reais. Toda a produção é centralizada em cenas, geralmente de duração de alguns segundos. Em contraste com simulações em tempo real, há pouca necessidade para a continuidade da simulação durante um longo período de tempo. É comum a distribuição da produção de uma cena em times de pessoas diferentes, que são depois compostos no processo de pós-produção.

O mais avançado sistema de animação de multidões para produções *offline* é *Massive*, utilizado em diversas superproduções cinematográficas, incluindo a trilogia *The Lord of the Rings*. Em *Massive*, cada agente faz escolhas a respeito de suas ações dependendo de suas entradas sensoriais utilizando um cérebro de centenas de nós lógicos [20]. De acordo com a decisão do cérebro, a movimentação a ser realizada é selecionada de uma extensa biblioteca de clipes de captura de movimento com transições pré-computadas. Por exemplo, no segundo filme da trilogia *The Lord of the Rings*, mais de 12 milhões de quadros de captura de movimento (equivalente a 55 horas de animação) foram utilizados. *Massive* também utiliza dinâmica de corpos rígidos, uma abordagem baseada em física para facilitar a animação de movimentações acrobáticas como cair, ou a animação de acessórios. Por exemplo, uma combinação de simulação baseada em física e clipes de captura de movimento personalizados

foi utilizada para criar a cena “*The Flooding of Isengard*”, onde *orcs* fogem de uma parede de água e caem em precipícios [36].



Figura 2.3: Screenshot do programa Massive, utilizado em produções cinematográficas.

Em comparação com aplicações em tempo real, a qualidade da movimentação e do visual em produções *offline* é muito superior, porém é obtida a grande custo. Por exemplo, para *The Lord of the Rings: The Two Towers*, a renderização de todos os personagens digitais demorou 10 meses para ser computada em uma potente fazenda de renderização com centenas de computadores [11].

### 2.3.6 Multidões em Jogos

Em jogos digitais atuais, multidões virtuais ainda são raras. O principal motivo é que multidões são custosas por natureza, não só em termos de recursos computacionais de tempo real, mas também em termos de custo de produção. Ainda assim, essa situação está começando a mudar, com o gênero de estratégia em tempo real (RTS) liderando o avanço, uma vez que o aumento no tamanho dos exércitos envolvidos tem um efeito direto no *gameplay*.

A principal preocupação em jogos é a velocidade na computação da renderização e do comportamento. Em comparação com produções *offline*, a qualidade de ambas a renderização e o comportamento são sacrificadas em troca de fluidez. Similarmente a produções cinematográficas, jogos digitais geralmente se utilizam de bibliotecas de animação as quais são em sua maioria obtidas por captura de movimento. A renderização utiliza técnicas de níveis de detalhe (*level-of-detail*, LOD), com alguns títulos empregando impostores animados.

Para melhorar os custos computacionais do comportamento em jogos que envolvem um grande número de entidades simuladas, LOD comportamentais são utilizados para baratear a computação. Em tais técnicas, o comportamento é computado somente para algumas entidades consideradas importantes (visíveis ou perto de se tornarem visíveis). No entanto, o manejo de LOD comportamentais é muito mais complexo do que o manejo de LOD na renderização. É perfeitamente correto não computar a visualização de agentes que não estão visíveis, porém não computar o comportamento desses agentes pode levar a uma simulação

incoerente. Em alguns jogos é comum o jogador causar alguma situação significativa (por exemplo, um congestionamento), então olhar para outro local, e quando voltar a olhar, a situação pode estar mudada em uma forma inesperada (o congestionamento é “magicamente” resolvido).

Em casos em que o cenário é composto de centenas ou milhares de unidades, muitas vezes a unidade selecionável com comportamento distinto é uma formação de tropas, não um soldado individual. O que parece ser muitas entidades na tela é na verdade uma entidade única sendo renderizada como várias partes separadas.

Reynolds [34] conseguiu a simulação de até 15,000 indivíduos em tempo real no Playstation 3. O modelo utiliza uma abordagem multi-agente baseada em regras, com uma técnica de *hashing* espacial, separando grupos de indivíduos e processando-os paralelamente. É importante levar em consideração que o conteúdo principal dos jogos digitais não é a simulação de multidões em si, alguns jogos podem apresentar multidões virtuais, porém necessitam recursos computacionais para computar outros elementos, impondo restrições ainda maiores sobre os recursos utilizados na simulação.

## **2.4 Multidões em mundos virtuais**

De forma a obter aplicações persuasivas utilizando multidões em ambientes virtuais, vários aspectos da simulação necessitam ser considerados, incluindo animação comportamental, modelagem do ambiente e renderização da multidão. Se não há uma renderização satisfatória, até o modelo comportamental mais complexo não será muito convincente. Caso o modelo comportamental não seja bom, mesmo que a simulação utilize um método de renderização perfeito, parecerá tola após somente alguns segundos. Se o ambiente não é modelado apropriadamente, os indivíduos não vão se comportar de forma acreditável, visto que eles vão realizar ações em locais errados ou até não realizá-las.

O objetivo da animação comportamental é facilitar o trabalho dos designers permitindo que personagens virtuais possam realizar independentemente ou semi-independentemente movimentações complexas as quais de outra forma requereriam grande quantidade de trabalho de animadores humanos; ou, em caso de aplicações interativas, os modelos comportamentais permitem que os personagens respondam a ações iniciadas pelo usuário.

Para o comportamento fazer sentido, além dos personagens, o ambiente também tem de ser modelado, não só graficamente, mas também semanticamente. De fato, um repertório de ações possíveis é dependente do que está incluído ou não está incluído na modelagem do ambiente. É muito comum o fato de o ambiente ser rico visualmente, porém as possíveis interações que os personagens podem realizar sejam mínimas.

Finalmente, para aplicações interativas, é necessário mostrar um conjunto de personagens distintos de forma eficiente. Personagens renderizados devem se encaixar visualmente no ambiente, eles devem ser afetados pela luz e outros efeitos da mesma forma como o ambiente.

A seguir, nós apresentaremos trabalhos representativos realizados em cada um desses tópicos agrupados em seu foco principal.



## 2.5 Comportamento e animação de grupos e multidões

O ser humano é indiscutivelmente a criatura conhecida mais complexa até então, portanto, são as mais complexas criaturas a serem simuladas. Uma animação comportamental de uma multidão composta de seres humanos (ou humanóides) é baseada em fundações da simulação de grupos de outras entidades mais simples, como bando de aves [32] e escolas de peixes [47].

Idealmente, um modelo para a simulação comportamental de multidões deve ser ao mesmo tempo eficiente e realista, ou seja, deve ser capaz de simular um grande número de indivíduos eficientemente de forma a causar um impacto mínimo nos recursos computacionais da aplicação. Na prática, no entanto, há geralmente uma troca entre realismo e eficiência, e, portanto, as características comportamentais dos modelos para a simulação de multidões variam muito dependendo dos objetivos da aplicação final.

Em seu trabalho pioneiro, Reynolds [32] descreveu um modelo comportamental distribuído para simular o movimento agregado de um bando de aves. A idéia revolucionária vinha do fato de que um comportamento complexo de um grupo de atores poderia ser obtido através de regras locais simples para os membros do grupo ao invés de uma condição global. O bando é simulado como um complexo sistema de partículas, com as aves simuladas (chamadas *boids*) sendo representadas através das partículas. O modelo de *boids* proposto consiste de três simples regras: (i) evitar colisão com outros *boids*, (ii) igualar a velocidade com *boids* vizinhos, e (iii) voar em direção ao centro do bando. Os resultados obtidos demonstraram que de regras locais simples pode emergir um comportamento muito mais complexo. Reynolds estendeu consecutivamente seu modelo em diversos trabalhos posteriores [33, 31], incorporando regras mais complexas e permitindo a simulação de até 15,000 indivíduos em tempo real [34].



Figura 2.4: Peixes artificiais apresentados por Tu e Terzopoulos [47].

Tu e Terzopoulos propuseram um framework para a animação de peixes artificiais [47]. A complexidade da vida marinha, incluindo interações entre peixes como predadores caçando presas, acasalamento, e formação de escolas, foi obtida através da modelagem individual do comportamento: o comportamento coletivo emergiu a partir da interação dos indivíduos. Neste sentido, este trabalho é similar ao trabalho de Reynolds, porém apresentando complexidade superior. O estado interno dos agentes inclui fome, libido, medo, e também alguns parâmetros de hábito, indicando, por exemplo, se preferem água fria, morna ou

quente. As intenções dos peixes são geradas por um algoritmo que se utiliza desses aspectos (em adição aos sentidos) para produzir uma saída. A execução final das intenções geradas é então realizada através de “rotinas de comportamento”, que especificam qual ação baixo-nível deve ser executada para satisfazer as intenções do peixe.

Brogan e Hodgings [15, 4] simularam o comportamento coletivo para sistemas com “dinâmica significativa”. Comparado aos *boids*, uma movimentação mais realista é obtida levando-se em conta propriedades físicas do movimento, tais como momento ou equilíbrio. Um dos estudos de caso apresentados simula a dinâmica de robôs de uma perna, os quais se movem pulando. Esses robôs apresentam “dinâmica significativa” porque eles não podem mudar intencionalmente sua velocidade durante o pulo, enquanto não estiverem no chão. Além dos robôs, os autores apresentaram resultados de simulações envolvendo ciclistas modelados através de uma hierarquia de corpos físicos conectados por articulações.

Musse e Thalmann [26, 25] apresentaram um modelo hierárquico para a simulação em tempo real de multidões humanas virtuais. O modelo proposto é baseado em hierarquia de grupos: grupos são estruturas mais inteligentes, onde os indivíduos seguem as especificações do grupo. Grupos podem ser controlados por diferentes níveis de autonomia: multidões guiadas seguem ordens (como ir para um determinado local) dadas pelo usuário em tempo real; multidões programadas seguem um comportamento prescrito; e multidões autônomas utilizam eventos e reações para criar comportamentos complexos. O ambiente é composto de pontos de interesse, os quais significam objetivos e referências de trajetória; e de pontos de ações, os quais são objetivos com algumas ações associadas. Agentes se movem entre os pontos seguindo curvas de Bezier.

Outro trabalho que se baseia em hierarquia é o proposto por Niederberger e Gross [27], o qual utiliza uma arquitetura de agentes hierárquicos e heterogêneos. Comportamentos são definidos através da especialização de tipos de comportamento já existentes e por hierarquia múltipla com pesos para a criação de novos tipos. Grupos são definidos através de padrões recursivos ou de padrões baseados em módulo. O motor de comportamento permite a especificação de um tempo máximo de computação, de forma a manter um custo mínimo e constante.

Ulicny e Thalmann [48, 49] apresentaram um simulador comportamental de multidões com uma arquitetura modular para um sistema multi-agente permitindo comportamentos autônomos e prescritos, suportando variedade. O comportamento é computado em camadas, onde decisões são tomadas por regras de comportamento e a execução é tratada por máquinas-de-estado hierárquicas.

Utilizando uma abordagem de dinâmica contínua, Treuille et al. [39] propôs um modelo no qual um campo de potencial dinâmico integra a navegação global com obstáculos móveis, resolvendo eficientemente o problema de locomoção de grandes multidões sem a necessidade explícita de tratamento de colisão.

A complexidade percebida durante a simulação de multidões pode ser aumentada utilizando níveis de detalhe (LOD). O’Sullivan et al. [28] descreveu a simulação de multidões e grupos de indivíduos utilizando níveis de detalhe para geometria, locomoção e comportamento. No nível

geométrico, técnicas de subdivisão são utilizadas para a obtenção de uma transição suave entre mudanças de LOD. No nível de locomoção, os movimentos são simulados utilizando LOD adaptativo. Para o comportamento, LOD é empregado para reduzir o custo computacional da atualização de indivíduos que são menos importantes. Indivíduos mais complexos se comportam de acordo com seus papéis e motivações, enquanto menos complexos simplesmente apresentam comportamento aleatório.

## 2.6 Renderização de multidões

Em um mundo ideal, as placas gráficas seriam capazes de, a cada quadro, renderizar um número infinito de triângulos com uma complexa e arbitrária iluminação. Para a visualização de multidões, nós então simplesmente utilizaríamos milhares de malhas detalhadas, capazes de permitir animação de rosto e de mão. Infelizmente, mesmo com o avanço do hardware gráfico, nós ainda somos limitados ao uso de uma quantidade finita de triângulos por quadro. Essa quantidade permitida tem de ser gasta sabiamente para a visualização de multidões densas sem muita degradação perceptível.

Outro problema inerente à visualização de multidões é a necessidade do gerenciamento da variedade dos indivíduos. Ao simular um pequeno grupo de humanos virtuais, é fácil fazê-los parecerem distintos individualmente, basta utilizar malhas e texturas distintas para cada humano presente na cena, assinalando-os animações diferentes. Porém, quando o grupo é entendido a uma multidão de milhares de pessoas, essa solução se torna impraticável. Primeiro, em termos de design, é inimaginável a criação de um modelo e uma série de animações para cada indivíduo. Segundo, a memória necessária para guardar todos os dados seria muito exigente. Infelizmente, não há uma solução direta a este problema. Existem, no entanto, técnicas hierárquicas que são capazes de produzir uma grande quantidade de variedades. Na Figura 2.5 é exemplificado este problema: além da geração de diversas pessoas visualmente distintas, é necessário distribuí-las de forma eficiente.



Figura 2.5: Geração de variedade utilizando modelos distintos e modulação de textura.

A renderização em tempo real de uma grande quantidade de personagens 3D é um desafio considerável; é capaz de exaurir os recursos do sistema rapidamente até para sistemas modernos com extensos recursos de memória, processadores rápidos, e placas gráficas potentes. Abordagens de “força bruta” que são normalmente utilizadas quando há poucos elementos não escalam quando existem centenas, milhares e até mais personagens. Diversos trabalhos tentam rodear essas limitações através de um uso inteligente dos recursos das placas gráficas e empregando métodos que lucram do fato de que nossa percepção da cena como um todo é limitada.

Nós só conseguimos perceber em detalhe somente uma quantidade relativamente pequena de uma grande coleção de personagens. Um simples cálculo mostra que tratar todos os membros da multidão igualmente é um desperdício. Telas modernas são capazes de mostrar cerca de dois milhões de pixels ao mesmo tempo, enquanto um personagem razoavelmente complexo é composto de cerca de 10,000 triângulos. Mesmo assumindo que todos os triângulos fossem projetados em um único pixel na tela, e que os personagens não se sobrepõem, a tela completamente cheia de personagens seria coberta por uma multidão contendo somente 200 indivíduos visíveis simultaneamente. Na verdade, o número real é menor, uma estimativa mais razoável é de cerca de algumas dúzias de indivíduos completamente visíveis, com o resto da multidão escondida atrás desses indivíduos ou tomando um espaço significativamente menor da tela. Dessa forma, faz sentido a utilização de aproximações menos complexas para os personagens menos visíveis. Técnicas de níveis de detalhes então trocam a visualização de acordo com a posição e orientação do observador. No trabalho de Hamil et al. [16] foram feitos testes em como a movimentação afeta a percepção de um ser humano representado ou por um impostor ou por uma estrutura geométrica, com o objetivo de definir as distâncias nas quais as trocas entre os modelos fossem minimamente notadas.

Impostores *billboard* são um dos métodos utilizados para acelerar a renderização de uma multidão. Impostores são polígonos com uma textura parcialmente transparente que contém uma foto capturada de um personagem 3D de alta definição e estão sempre voltados em direção à câmera. Aubel et al. [1] introduziu impostores gerados dinamicamente para a renderização de humanos virtuais animados. A abordagem utiliza um processo de criação de impostores rodando em paralelo a simulações completamente 3D, a qual tira fotos dos personagens 3D renderizados. Essas fotos em cache são então utilizadas no local da geometria durante vários quadros, até que um movimento suficiente da câmera ou do personagem gere a necessidade de outra foto, renovando a textura do impostor.

Em outro importante trabalho utilizando impostores, Tecchia et al. [42] propôs um método para a renderização em tempo real de uma multidão animada em uma cidade virtual. Comparado a métodos anteriores, impostores não são computados dinamicamente, mas são criados em um pré-processamento. Fotos são amostradas de pontos de visão distribuídos em uma esfera ao redor do personagem. Esse processo é repetido para todos os quadros da animação. Durante a visualização, as imagens obtidas do ponto de visão mais próximo à posição atual da câmera são utilizadas na textura do *billboard*. Adicionalmente, as silhuetas dos impostores são utilizadas para a geração de sombras projetadas na superfície do chão. Multitexturização é utilizada para adicionar variedade através da modulação das cores do impostor. Em um trabalho posterior, foi adicionada iluminação e mapas de normais [43]. O método utilizando impostores pré-computados é mais rápido do que impostores dinâmicos, porém demanda muito de memória de textura, o que leva a menor qualidade de imagem, visto que o tamanho das texturas tem de ser mantido pequeno.

Uma possibilidade distinta para a visualização rápida de uma multidão são as técnicas de renderização baseada em pontos. Wand e Strasser [53] apresentaram uma abordagem de renderização multi-resolução a qual unifica a renderização baseada em polígonos e a renderização baseada em imagens. É criada uma representação em octree dependente da visão para quadro de animação, onde os nós ou guardam um polígono ou um ponto. Essas

representações podem ser interpoladas linearmente entre uma árvore e outra de forma a calcular os quadros intermediários. Quando o observador está a uma longa distância, é utilizada renderização baseada em pontos, quando está próximo, é utilizada técnicas de polígono, e quando está no meio, uma mistura das duas.

## **2.7 Modelagem do ambiente para multidões**

A modelagem do ambiente é diretamente relacionada à animação comportamental. O principal objetivo do modelo do ambiente é facilitar a simulação das entidades que o habitam. A credibilidade de criaturas virtuais pode ser reforçada significativamente se elas se comportam de acordo com o ambiente em que estão inseridas, caso elas realizem uma ação não permitida ou inesperada, como por exemplo atravessar uma parede, a descrença será imediata. Os esforços são direcionados a representações e algoritmos que previnam comportamentos “proibidos” de acontecerem. Até recentemente, os dois maiores problemas da inteligência artificial em jogos digitais eram tratamento de colisão e planejamento de trajeto [8].

A maioria da população no mundo em que vivemos mora nas cidades; é nas cidades onde a maioria das atividades humanas acontece nos tempos atuais. Conseqüentemente, a maior parte das pesquisas é direcionada a modelagem de cidades virtuais. Farenc et al. [12] introduziu um ambiente informado dedicado à simulação de humanos virtuais em um contexto urbano. O ambiente informado é um banco de dados integrando informação semântica e geométrica a respeito da cidade virtual. É baseado em uma decomposição hierárquica de uma cena urbana em entidades do ambiente, como quarteirões, blocos, entroncamentos, ruas, e assim por diante. As entidades podem conter uma descrição dos comportamentos apropriados a serem realizados pelos agentes ocupantes; por exemplo, uma calçada informa que deve ser andada, ou um banco informa que deve ser sentado. Além disso, o banco de dados do ambiente pode ser utilizado para a busca de trajeto que é personalizado ao tipo de cliente, de forma a, por exemplo, um pedestre receber um caminho utilizando calçadas, porém um carro deve receber um caminho utilizando estradas.

Outro modelo de cidade virtual para a animação do comportamento foi apresentado por Thomas e Donikian [40]. O modelo foi criado com ênfase na simulação de trânsito de veículos e pedestres. O ambiente é um banco de dados dividido em duas partes: uma estrutura hierárquica contendo uma árvore de regiões poligonais, similar ao ambiente informado; e uma estrutura topológica com um grafo da rede de estradas. As regiões contêm informações sobre direções de circulação, incluindo possíveis mudanças de rota nas interseções. Os agentes então utilizam o banco de dados para navegar pela cidade.

Sung et al. [37] apresentou uma nova abordagem para controlar o comportamento de uma multidão guardando informação comportamental dentro do ambiente utilizando estruturas chamadas “situações”. Comparada às abordagens anteriores, estruturas ambientais (situações) podem se sobrepor; comportamentos nessas áreas de sobreposição são então compostos utilizando distribuições de probabilidade. Funções de comportamento definem probabilidade de transições de estado (causando animações de movimento) dependendo do estado dos recursos do ambiente ou do estado anterior do agente.

Com foco em problemas mais genéricos de planejamento de trajeto, diversos trabalhos foram realizados. Kallmann et al. [19] propôs um algoritmo de planejamento de trajeto rápido baseado na triangulação de Delaunay restrita completamente dinâmica. Bayazit et al. [5] utilizou mapas de ruas globais para melhorar o comportamento em grupo em ambientes geometricamente complexos. Grupos de criaturas exibem comportamentos tais como retorno à casa, busca de objetivo, cobertura, ou liderança, através de regras embutidas nos membros do grupo e nos mapas de ruas. Lamarche e Donikian [22] apresentaram uma estrutura topológica do ambiente geométrico para um planejamento de trajeto rápido e hierárquico e um algoritmo de navegação reativa para multidões virtuais. Pettré et al. [30] apresentou como computar de forma automática e robusta um grafo de navegação multi-nível utilizando cilindros tridimensionais.

### 3 Implementação

Diversos são os modelos existentes para a simulação do comportamento de multidões, utilizando abordagens microscópicas, onde cada indivíduo possui um comportamento e objetivo individual e interage com outros indivíduos, ou abordagens macroscópicas, onde a multidão é simulada como um todo (características do fluxo ao invés de individual). Para nossa implementação, decidimos pelo modelo proposto por Treuille et al. [39], um modelo macroscópico que integra em tempo real a navegação global com pessoas e obstáculos se movendo através de um campo de potencial dinâmico.

O modelo *continuum crowds* é um modelo em tempo real para a simulação da movimentação de milhares de pessoas ao mesmo tempo. Ele se mostra bastante promissor para a integração de multidões em aplicações iterativas como jogos, e, mesmo sendo um modelo macroscópico, permite a utilização integrada de agentes autônomos de forma simples e eficiente. Sua implementação é não trivial, fundamentada matematicamente, mais especificamente na dinâmica de fluidos, mostrando-se um ótimo modelo para nossa implementação, atingindo os objetivos de nossa proposta.



Figura 3.1: Resultado do modelo *continuum crowds* apresentado por Treuille et al [39].

A linguagem escolhida para o desenvolvimento do projeto foi *Java*. Apesar de apresentar um desempenho inferior (principalmente a nível de utilização de memória) comparada a uma linguagem totalmente compilada como *C* ou *C++*, *Java* é uma linguagem muito mais produtiva e “segura”. Erros típicos de aplicações matemáticas como divisão por zero, manipulação de valores infinitos, raiz quadrada de números negativos, e muitos outros, podem ser facilmente encontrados e eliminados; em outra linguagem com menos suporte a tais operações, o erro poderia ser mascarado e resultados incompreensíveis poderiam surgir.

O foco de nosso trabalho é voltado à animação comportamental da multidão. O problema de visualização não é abordado em nossa implementação, nós apresentamos um visualizador simples 2D.

### 3.1 Abordagem do comportamento

O trabalho de Treuille et al. [39] apresenta um modelo em tempo real que sintetiza a movimentação de grandes multidões sem a necessidade de dinâmicas baseadas em agentes, chamado *continuum crowds*. A movimentação da multidão é vista como uma minimização de energia entre partículas, e é adotada uma perspectiva contínua no sistema. Essa formulação produz um conjunto de campos de potencial dinâmico e de velocidade sobre o domínio que guiam toda a movimentação individual simultaneamente. A abordagem unifica planejamento de trajeto global e tratamento de colisão local em um único framework de otimização. As pessoas não sofrem uma troca discreta em seu regime na presença de outras pessoas, elas realizam planejamento global para evitar obstáculos e outros indivíduos. A formulação do campo de potencial dinâmico também garante que os trajetos são ótimos para o estado atual do ambiente, impedindo que os indivíduos fiquem presos em mínimos locais.

É importante constatar que planejamento global de trajetória é frequentemente uma suposição irrealista. Normalmente, as pessoas possuem visão do ambiente limitada e conhecimento parcial do terreno. Conhecimento global é só uma aproximação de um planejamento de longo prazo com conhecimento e visão limitados. Ainda assim, segundo eles, a hipótese de planejamento global produz uma movimentação de multidão muito mais realista e suave do que a hipótese igualmente irrealista de planejamento com conhecimento estritamente local. Ao mesmo tempo, o framework não é tão geral quanto métodos baseados em agentes: há uma troca da variedade individual pelo planejamento do comportamento ótimo de uma multidão em tempo real com uma computação mínima por indivíduo. Para alcançar os benefícios de ambos os modelos é possível a integração de agentes ao *continuum crowds* sem dificuldades.

O sistema utiliza duas simples idéias as quais são cruciais para a criação dos efeitos resultantes: uma é um termo dependente da velocidade, o qual induz a formação de raia, e a outra é um termo baseado na distância, o qual estabiliza o fluxo. Também é apresentado como indivíduos podem integrar conhecimento do futuro para produzir um comportamento mais inteligente. O modelo exhibe importantes fenômenos emergentes observados em multidões reais; em particular, pessoas andando em direções diferentes tendem a formar raia, e grupos se cruzando podem formar vórtices.

O custo computacional do algoritmo depende do número de células de grade utilizado para computar o potencial dinâmico. Em discretizações grosseiras, é possível a simulação de mais de dez mil pessoas a vários quadros por segundo, enquanto em discretizações finas, é possível a simulação em alta resolução da dinâmica de centenas de pessoas em tempo real.

A seguir, apresentamos um resumo da teoria do modelo *continuum crowds*. Nós fortemente recomendamos o leitor a investigar o artigo original [39] para maior entendimento do modelo e das derivações das funções apresentadas.

#### 3.1.1 O Modelo

Através de uma série de hipóteses de como as pessoas escolhem seus caminhos, é formulada a função de custo de um caminho. Em geral, as pessoas escolhem o caminho com distância mínima ao seu destino. No entanto, essa preferência é moderada pelo desejo de evitar



congestionamento e outras situações consumidoras de tempo. Isso pode ser visto como a clássica troca entre a minimização do tempo e da energia. Adicionalmente, as pessoas preferem minimizar sua exposição a áreas de alto desconforto. Essas idéias são então resumidas assumindo que as pessoas escolhem seus trajetos de forma a minimizar uma combinação linear dos seguintes termos:

- O tamanho do trajeto.
- A quantia de tempo até o destino.
- O desconforto sentido, por unidade de tempo, durante a trajetória.

Assumindo que o campo de velocidade  $f$ , desconforto  $g$ , e objetivos  $G$  são fixos, uma pessoa localizada em  $\mathbf{x}$  escolherá o caminho  $P$  de forma a minimizar:

$$\int_P C ds, \text{ onde } C \equiv \frac{\alpha f + \beta + \gamma g}{f}, \quad (1)$$

onde  $\alpha$  (tamanho do trajeto),  $\beta$  (tempo),  $\gamma$  (desconforto) são pesos dos termos individuais;  $ds$  significa que a integral é calculada com relação ao tamanho do trajeto, e  $C$  é o *campo de custo unitário*.

A trajetória ótima é então encontrada através de uma função de potencial utilizando o custo do trajeto descrito pela Equação (1). Suponha que temos uma função  $\Phi: \mathbf{R}^2 \rightarrow \mathbf{R}$  em todo local igual ao custo do trajeto ótimo ao objetivo. Intuitivamente, faz sentido que, para qualquer pessoa, a estratégia ótima é se mover ao oposto do gradiente desta função, uma vez que isso irá diminuir o custo do trajeto mais rapidamente. Reciprocamente, essa função pode ser construída seguindo-se o conjunto de todos os trajetos ótimos partindo-se do objetivo em direção ao exterior, integrando o custo ao longo do caminho. De fato, a função potencial  $\Phi$  pode ser definida exatamente desta forma: no objetivo  $\Phi = 0$ , e em todos os outros locais  $\Phi$  satisfaz a *equação eikonal*:

$$||\nabla\Phi(\mathbf{x})|| = C, \quad (2)$$

onde o custo unitário  $C$  é avaliado na direção do gradiente  $\nabla\Phi$ . No sistema, todas as pessoas se movem na direção oposta ao gradiente, escalada pela velocidade naquele ponto:

$$\dot{\mathbf{x}} = -f(\mathbf{x}, \theta) \frac{\nabla\Phi(\mathbf{x})}{||\nabla\Phi(\mathbf{x})||}, \quad (3)$$

onde  $\dot{\mathbf{x}}$  denota a velocidade final, e  $f(\mathbf{x}, \theta)$  é calculado na direção do movimento.

O campo de velocidade  $f$  mede a máxima velocidade de movimento permitida para cada ponto e para cada direção no domínio. A velocidade é uma variável dependente da densidade: em baixas densidades, a velocidade é dominada pelo terreno, se mantendo constante em superfícies planas, porém mudando com a inclinação; em altas densidades, a velocidade é dominada pelo movimento das pessoas próximas: o movimento é inibido na direção contrária ao fluxo, porém não é afetado quando em direção ao fluxo.

Como a velocidade é dependente da densidade, é necessária a utilização de um *campo de densidade*  $\rho$ . Cada pessoa é convertida em um campo individual de densidade, denotado  $\rho_i$

para a i-ésima pessoa. Esse campo deve ter seu pico na localização da i-ésima pessoa e declinar radialmente. A densidade da multidão  $\rho$  é simplesmente a soma de cada campo de densidade individual. Ao computar o campo de densidade, o *campo de velocidade média*  $\bar{\mathbf{v}}$  deve ser simultaneamente calculado, o qual escala a densidade de cada indivíduo por sua velocidade, indicando a velocidade e direção geral do fluxo da multidão:

$$\rho = \sum_i \rho_i, \quad \text{e} \quad \bar{\mathbf{v}} = \frac{\sum_i \rho_i \mathbf{\dot{x}}_i}{\rho}, \quad (4)$$

onde  $\mathbf{\dot{x}}_i$  denota a velocidade da i-ésima pessoa. Ambas as somas são realizadas através de todas as pessoas em todos os grupos.

O campo de velocidade  $f$  é calculado levando em consideração dois fatores: em áreas de baixa densidade ( $\rho \leq \rho_{min}$  para algum  $\rho_{min}$ ) a velocidade  $f$  é igual a *velocidade topográfica*  $f_T$ . Em áreas de alta densidade ( $\rho \geq \rho_{max}$  para algum  $\rho_{max}$ ) a velocidade  $f$  é igual a *velocidade de fluxo*  $f_{\bar{\mathbf{v}}}$ . Quando a densidade está entre  $\rho_{min}$  e  $\rho_{max}$ , é realizada uma interpolação linear entre os dois valores.

Assumindo que o terreno está limitado às inclinações mínima e máxima  $s_{min}$  e  $s_{max}$ , a velocidade topográfica é dada por:

$$f_T(\mathbf{x}, \theta) = f_{max} + \left( \frac{\nabla h(\mathbf{x}) \cdot \mathbf{n}_\theta - s_{min}}{s_{max} - s_{min}} \right) (f_{min} - f_{max}), \quad (5)$$

onde  $\nabla h(\mathbf{x}) \cdot \mathbf{n}_\theta$  é a inclinação do campo de altura  $h$  em direção a  $\theta$  e  $f_{min}$  e  $f_{max}$  são as velocidades mínima e máxima que a pessoa pode atingir.

A velocidade de fluxo  $f_{\bar{\mathbf{v}}}$  é essencialmente a velocidade média  $\bar{\mathbf{v}}$  calculada a uma distância  $r$  da localização  $\mathbf{x}$ :

$$f_{\bar{\mathbf{v}}}(\mathbf{x}, \theta) = \bar{\mathbf{v}}(\mathbf{x} + r\mathbf{n}_\theta) \cdot \mathbf{n}_\theta. \quad (6)$$

O deslocamento faz com que as pessoas avaliem a velocidade das áreas as quais elas estão tentando alcançar. De fato, se não pelo deslocamento, a velocidade de uma pessoa seria dominada por sua própria velocidade anterior, um efeito não desejado. Além disso, a velocidade de fluxo é limitada para ser não negativa, significando que a multidão pode atrasar as pessoas, mas nunca carregá-las para trás.

Finalmente, o campo de velocidade  $f$  pode ser construído por:

$$f(\mathbf{x}, \theta) = f_T(\mathbf{x}, \theta) + \left( \frac{\rho(\mathbf{x} + r\mathbf{n}_\theta) - \rho_{min}}{\rho_{max} - \rho_{min}} \right) (f_{\bar{\mathbf{v}}}(\mathbf{x}, \theta) - f_T(\mathbf{x}, \theta)). \quad (7)$$

Note que a densidade é avaliada também com um deslocamento da localização atual, pelo mesmo motivo: não é desejado que uma pessoa contribua para a obstrução de sua própria movimentação.

O cálculo do campo potencial é custoso, porém um grupo de pessoas divide o mesmo objetivo, campo potencial e campo de velocidade. Esse geralmente é o caso em multidões, onde um número de pessoas tenta chegar ao mesmo destino com aproximadamente uma mesma

velocidade. Nesse caso, só precisamos calcular a função potencial para cada grupo uma única vez, derivando o trajeto ótimo para todos os membros do grupo simultaneamente. Na realidade, as pessoas se movem a diferentes velocidades, percebem o desconforto de forma distinta, e procuram objetivos diferentes; portanto, a multidão é dividida em um conjunto de grupos, cada um com características distintas. A cada passo de tempo é construída a função potencial  $\Phi$  para cada grupo, e então todas as pessoas se movimentam de acordo com a Equação (3). Como cada grupo afeta a velocidade  $f$  e desconforto  $g$  para todos os grupos, o movimento de toda a multidão é acoplado.

Resolver a equação (2) é o aspecto mais custoso da simulação. Portanto, é preferível ter poucos grupos. Contudo, mesmo somente um único grupo exibe dinâmicas interessantes, e congestionamentos realistas apresentados em multidões reais podem ser obtidos com poucos grupos.

### 3.1.2 Implementando o modelo

Para simular o sistema, o modelo descrito na ultima seção precisa ser discretizado no tempo e no espaço. A Figura 3.4 ilustra visualmente a visão geral do funcionamento do algoritmo. O simulador avança a cada passo de tempo da seguinte forma:

*Para cada passo de tempo:*

- Converte a multidão em um campo de densidade.

*Para cada grupo:*

- Constrói o campo de custo unitário  $C$ .
- Constrói o potencial  $\Phi$  e seu gradiente  $\nabla\Phi$ .
- Atualiza a localização das pessoas.
- Garante a distância mínima entre as pessoas.

Para computar esses campos, o espaço é discretizado em uma grade regular, com variáveis físicas definidas em vários locais dentro de cada célula de grade. O artigo original guarda todos os campos físicos em arrays 2D de números de ponto flutuante, seguindo o esquema mostrado na Figura 3.2. Todos os campos escalares são guardados no centro de cada célula de grade. A velocidade média também é guardada no centro de cada célula como um par de pontos flutuantes. Todos os campos anisotrópicos – aqueles dependentes de ambas posições e direções – são guardados com quatro pontos flutuantes por célula, correspondendo às faces leste, oeste, sul e norte. Finalmente, a velocidade final  $\mathbf{v}$ , o gradiente da altura  $\nabla h$  e o gradiente do potencial  $\nabla\Phi$  são guardados em um estilo de arrumação MAC.

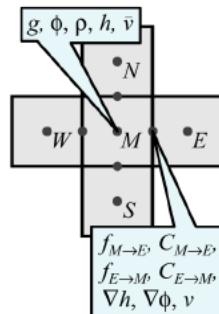


Figura 3.2: Estrutura de grade discretizada

Nós guardamos os campos de forma parecida com a deles, porém com algumas diferenças. Os campos físicos são guardadas em *arrays* 2D de números *double* (pontos flutuantes de maior precisão). Cada campo escalar no centro de uma célula ( $g, \Phi, \rho, h$ ) é guardado como um único *array*, enquanto os campos da face ( $f, C, \nabla h$ ) são guardados como quatro *arrays*, um correspondendo o valor em cada direção. A velocidade média  $\bar{v}$  e a velocidade final  $\mathbf{v}$  são guardados como um par de *arrays*. Não verificamos necessidade de guardar o gradiente do potencial  $\nabla\Phi$ . Nós não utilizamos o estilo MAC indicado pelo modelo, pois não o encontramos na literatura.

### 3.1.2.1 Conversão da densidade

O primeiro passo do simulador é converter a multidão em um campo de densidade. Como o espaço é discretizado, primeiro temos que converter a posição (contínua). Nós encontramos a célula mais próxima as quais ambas as coordenadas são menores do que as da pessoa. Nós então calculamos as coordenadas relativas  $[\Delta x, \Delta y]$  da pessoa em relação ao centro da célula como mostrado na Figura 3.3. A densidade da pessoa é então adicionada à grade na seguinte forma:

$$\begin{aligned}\rho_A &= \min(1 - \Delta x, 1 - \Delta y)^\lambda & \rho_B &= \min(\Delta x, 1 - \Delta y)^\lambda \\ \rho_C &= \min(\Delta x, \Delta y)^\lambda & \rho_D &= \min(1 - \Delta x, \Delta y)^\lambda,\end{aligned}$$

onde o *expoente de densidade*  $\lambda$  determina a velocidade da queda da densidade. Cada pessoa contribui pelo menos  $\bar{\rho}$  para sua célula de grade, mas não mais que  $\bar{\rho}$  para as células vizinhas, com  $\bar{\rho} = 1/2^\lambda$ . Ao computar o campo de densidade  $\rho$ , nós simultaneamente calculamos o campo de velocidade média  $\bar{v}$  de acordo com a Equação (4). Ao calcular o campo de densidade e o campo de velocidade média, é preciso ter cuidado quando as pessoas estão nas células limites da grade, devendo ignorar as células fora dos limites.

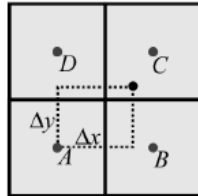


Figura 3.3: Estrutura do cálculo da densidade

### 3.1.2.2 Custo unitário

Existem dois passos para o cálculo do campo de custo unitário  $C$ . Primeiramente nós computamos o campo de velocidade  $f$  de acordo com a Equação (7), então nós calculamos o campo de custo  $C$  utilizando a Equação (1). Esses campos são anisotrópicos, portanto para computá-los, temos que iterar não só por todas as células de grade, mas também em todas as quatro direções em cada célula. Por exemplo, para a célula  $M$  na Figura 3.2, nós precisamos computar  $f_{M \rightarrow i}$  e  $C_{M \rightarrow i}$  para  $i \in \{\text{East}, \text{North}, \text{West}, \text{South}\}$ . Para que as pessoas antecipem os obstáculos em sua frente, nós devemos avaliar a velocidade e o desconforto na célula nas quais elas estão tentando se mover. Por exemplo, para o cálculo de  $f_{M \rightarrow E}$  e  $C_{M \rightarrow E}$ , nós utilizaríamos a densidade  $\rho_E$ , desconforto  $g_E$ , e velocidade média  $\bar{v}_E$  nas Equações (1) e (7).

### 3.1.2.3 Construindo o campo de potencial dinâmico

A construção do potencial dinâmico é o passo mais complexo e consumidor de tempo do modelo. A Equação (2) define o potencial como uma equação *eikonal* implícita, e portanto, não pode ser calculado diretamente. Porém, métodos eficientes foram desenvolvidos para a resolução deste tipo de equação, mais notavelmente, o *fast marching method* (FMM) [45].

O FMM é um método baseado no algoritmo de Dijkstra, ele trabalha sobre uma grade. A filosofia do método é trabalhar expandindo uma condição inicial. Inicialmente, todas as células são marcadas como *desconhecidas*, e recebem potencial  $\Phi = \infty$ . As células que formam a condição inicial (objetivo  $G$ , onde  $\Phi = 0$ ) são marcadas como *conhecidas*, e nós aproximamos os valores de seus vizinhos resolvendo uma aproximação de diferença finita da Equação (2). As células que possuem um valor computado, mas ainda não estão marcadas como conhecidas, são chamadas *candidatas*. Para cada iteração do laço central do algoritmo, a célula candidata com o menor potencial é marcada como conhecida, e suas vizinhas tem seu potencial re-approximado. Células conhecidas são utilizadas para calcular o valor de outras células, porém nunca são computadas novamente. Esse processo é repetido, propagando as células conhecidas do objetivo para fora até que todas as células sejam definidas. Uma estrutura de *heap* binário eficientemente cuida da lista de células candidatas e dá ao algoritmo o tempo de  $O(N \log N)$ , onde  $N$  é o número de células de grade.

A aproximação de diferença finita utilizada para aproximar a Equação (2) agora é descrita. Suponha que estamos resolvendo a equação para a célula  $M$  na Figura 3.2. Nós primeiro encontramos a célula adjacente de menor custo em ambos os eixos  $x$  e  $y$ :

$$m_x = \underset{i \in \{W, E\}}{\operatorname{argmin}} \{ \Phi_i + C_{M \rightarrow i} \} \quad m_y = \underset{i \in \{N, S\}}{\operatorname{argmin}} \{ \Phi_i + C_{M \rightarrow i} \},$$

onde *argmin* retorna o índice da célula vizinha de menor custo. Nós então utilizamos essas direções para calcular uma aproximação de diferença finita para a Equação (2) através da solução de maior valor para  $\Phi_M$  na equação quadrática:

$$\left( \frac{\Phi_M - \Phi_{m_x}}{C_{M \rightarrow m_x}} \right)^2 + \left( \frac{\Phi_M - \Phi_{m_y}}{C_{M \rightarrow m_y}} \right)^2 = 1. \quad (8)$$

Se  $m_x$  ou  $m_y$  é indefinido porque ambos os vizinhos tem custo infinito ou são desconhecidos, nós removemos esta dimensão da Equação (8). Note que até a discretização dos ângulos, essa formulação lida corretamente com a anisotropia do campo de custo unitário  $C$ .

Uma vez que nós tivermos computado  $\Phi_M$ , nós tiramos a diferença com as células de grade vizinhas  $(m_x, m_y)$  nos dando o gradiente  $\nabla \Phi$ . Nós então normalizamos o gradiente e o multiplicamos pelas velocidades nas direções apropriadas para calcular o campo de velocidade final  $\mathbf{v}$  naquele local.

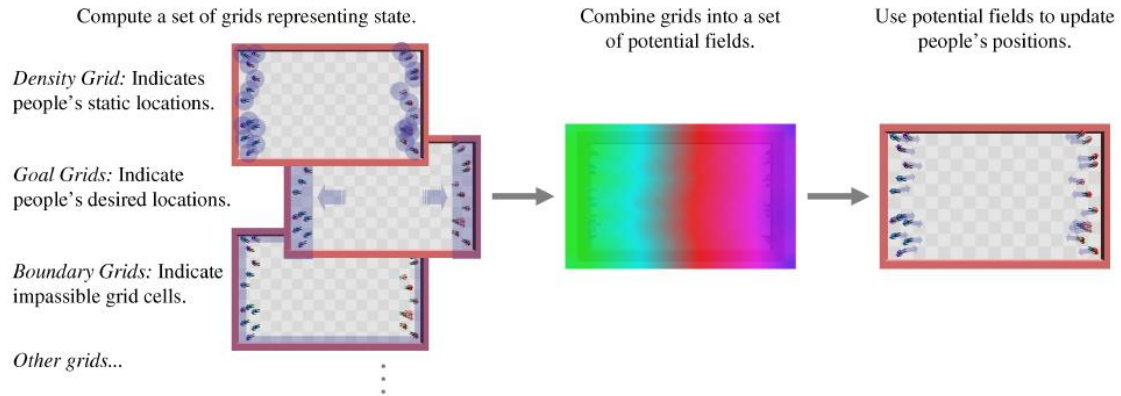


Figura 3.4: Visão geral do algoritmo apresentada no artigo original

### 3.1.2.4 Movimentando a multidão

Após calcular o campo potencial  $\Phi$ , seu gradiente  $\nabla\Phi$ , e determinado o campo de velocidade final  $\mathbf{v}$ , nós então simplesmente atualizamos as posições de cada pessoa de acordo com uma interpolação bilinear do campo  $\mathbf{v}$  (os valores das quatro células mais próximas são interpolados), efetivamente computando uma integração de Euler da Equação (3).

### 3.1.2.5 Garantindo a distância entre as pessoas

Em teoria, o modelo garante que duas pessoas não se intersectam. Uma pessoa se aproximando de outra irá eventualmente experimentar uma densidade tão grande que o termo da velocidade média irá dominar a equação da velocidade e a velocidade em direção à interseção irá cair para zero. Na prática, porém, só é possível resolver a dinâmica até a resolução da grade, e duas pessoas na mesma célula de grade irão eventualmente se intersectar. Quando essas intersecções acontecem, elas geram artefatos visuais desagradáveis.

Para resolver esse problema, é necessário garantir uma distância mínima entre as pessoas. Para tal, nós simplesmente temos que iterar sobre todos os pares de pessoas que estão com uma distância menor que um limitante, simetricamente as separando para que a distância mínima seja garantida. Para acelerar esse passo do algoritmo, nós associamos cada pessoa à sua célula de grade e só realizamos esse teste com as células vizinhas.

### 3.1.3 Estrutura de classes

Analisando o modelo *continuum crowds*, é possível identificar três principais elementos: o ambiente, os grupos de multidões, e as pessoas. Em nossa implementação, nós seguimos esses elementos, dividindo-os em duas classes cada, uma contendo as informações contínuas, e outra contendo as informações discretizadas. Dessa forma, a lógica principal do modelo é composta de quatro classes: **World**, que contém o ambiente contínuo, uma lista dos grupos, e um ponteiro para a discretização do ambiente; **WorldDiscretization**, que representa a discretização do ambiente, contendo os campos de desconforto ( $g$ ), de altura ( $h$ ), de densidade ( $\rho$ ), de velocidade média ( $\bar{v}$ ) e o campo anisotrópico do gradiente da altura ( $\nabla h$ ); **Crowd**, que representa um grupo de multidão e contém uma lista de membros (pessoas) e um ponteiro para a discretização da multidão; e **CrowdDiscretization**, que representa a discretização de um grupo de multidão, contendo os campos relativos à cada grupo.

Além dessas quatro classes, nós temos uma estrutura de pessoa, **Person**, a qual possui como atributos a posição contínua, velocidade, orientação e parâmetros de animação; uma classe utilitária chamada **ContinuumMath**, que contém funções utilitárias para o cálculo das funções do modelo; uma estrutura **Matrix** que convenientemente guarda um array 2D de *doubles*, que é como todos os campos de grade são representados; e um *heap* binário, chamado **BinaryHeap**, que é utilizado para a construção do potencial dinâmico como descrito na seção anterior.

### 3.2 Abordagem da renderização

O foco de nosso trabalho é voltado à animação comportamental dos indivíduos. Portanto, nós apresentamos um simples visualizador 2D. Na visualização, nós mostramos o ambiente composto pela grade discretizada, e as pessoas como sendo representados por um círculo, com a cor representando o grupo do qual ela faz parte.

### 3.3 Abordagem do ambiente

Como nós trabalhamos com uma discretização do ambiente e apresentamos um simples visualizador 2D, nós decidimos modelar o ambiente diretamente utilizando uma grade discretizada. Dessa maneira, as informações do ambiente como obstáculos e afins são representadas por grades, da mesma maneira como são representados os campos.

## 4 Resultados Obtidos

Assim como o artigo original, nós rodamos uma série de simulações com nossa implementação e confirmamos que o modelo *continuum crowds* pode produzir comportamento suave para milhares de agentes em tempo real. Todas as simulações foram rodadas em um Pentium Core 2 Duo de 2GHz. Nós encorajamos o leitor a visualizar os resultados de nossas simulações no vídeo anexado a este trabalho.

De forma a comparar nossa implementação com a implementação apresentada no artigo original, nós tentamos simular de forma igual dois cenários apresentados. A Figura 4.1 mostra a simulação original de vinte e quatro pessoas atravessando um largo corredor. É notável a formação de raia para a prevenção da colisão entre os grupos opostos. Esse fenômeno é ubíquo em multidões reais, e pode ser observado em todas as simulações, ambos no artigo original e em nossa implementação. A Figura 4.2 mostra nosso resultado deste mesmo cenário: ignorando as diferenças do visualizador, fica claro que nosso trabalho apresenta resultados bastante semelhantes. Neste cenário, devido à grade ser pequena e só apresentar dois grupos, nossa simulação rodou a mais de 100fps (quadros por segundo).



Figura 4.1: Dois grupos atravessam um corredor, naturalmente formando raia: resultado apresentado no artigo *continuum crowds*.

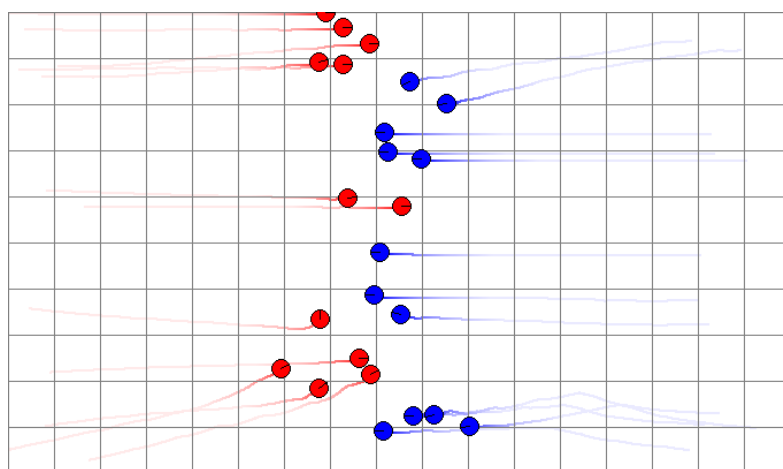
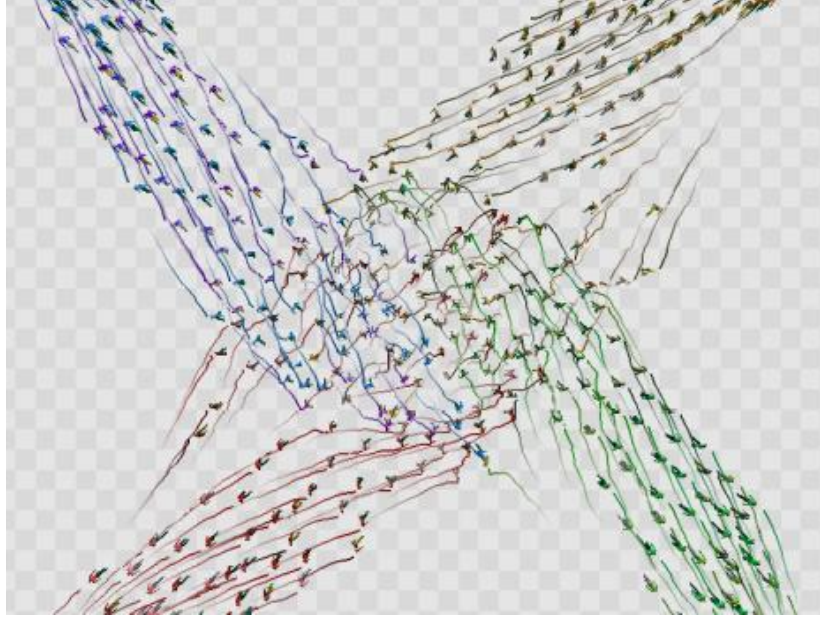


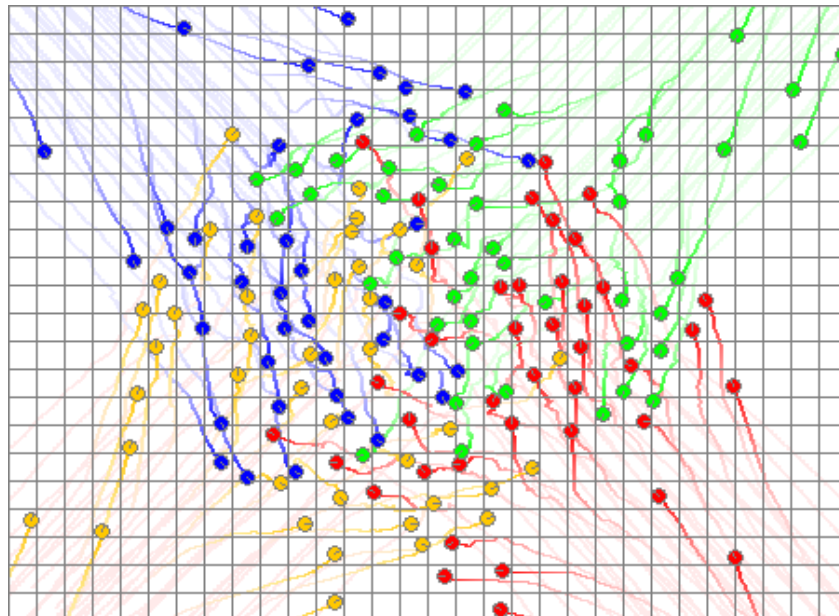
Figura 4.2: Nosso resultado de um cenário similar, apresentando corretamente o mesmo fenômeno de formação de raia.



Na Figura 4.3 é apresentado outro cenário do artigo original, onde quatro grupos diferentes se atravessam para alcançar os cantos opostos do domínio. Um padrão de vórtex se forma quando os grupos se cruzam. Esse comportamento emergente é uma forma exótica de formação de raias, e também serve para evitar colisões. Novamente, na Figura 4.4 nós apresentamos o nosso resultado para essa mesma simulação, e mais uma vez nós vemos que nossos resultados se assemelham aos apresentados originalmente. Neste cenário, nossa simulação rodou a uma média de 55fps.

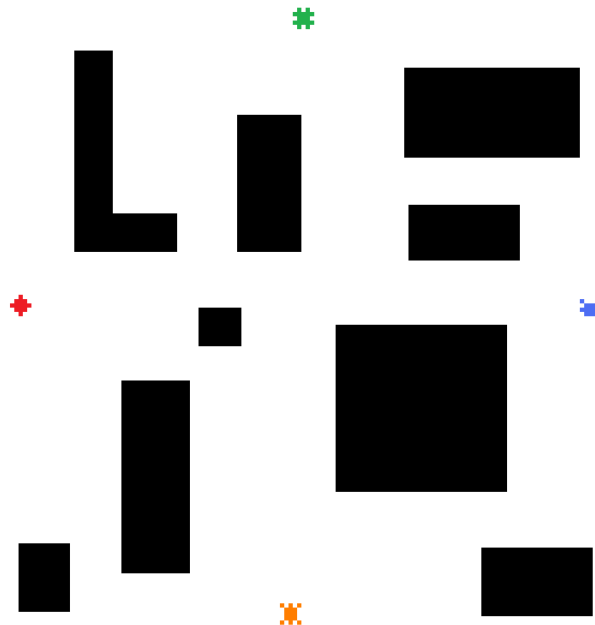


**Figura 4.3:** Quatro grupos se cruzam, formando um padrão de vórtex: resultado apresentado no artigo *continuum crowds*.



**Figura 4.4:** Nosso resultado de um cenário similar, apresentando corretamente o mesmo fenômeno de formação de padrões de vórtex.

Nós criamos um cenário que não foi baseado no artigo original. Um ambiente quadrado, com diversos obstáculos, inclui quatro grupos de pessoas, cada um com objetivo em um local do cenário, como indicado pela Figura 4.5. Quando uma pessoa chega ao seu objetivo, ela é inserida em outro grupo, adquirindo outro objetivo. Isso permite uma simulação por tempo indefinido e serve para mostrar todas as características apresentadas pelo modelo. Em diversos locais é apresentado o fenômeno de formações de raias, e em cruzamentos emerge o padrão de vórtex. Cada grupo possui inicialmente cerca de três mil pessoas, totalizando mais de dez mil pessoas. Neste tipo de cenário, nossa simulação roda a uma média de 5fps.



**Figura 4.5: Exemplo de cenário criado por nós: cada objetivo é representado por uma cor. Os obstáculos estão representados em preto.**

## 5 Considerações finais

Pode-se dizer que os objetivos traçados para este trabalho foram atingidos. Inicialmente nós apresentamos o estado da arte relativo à simulação de multidões em todas as áreas abrangidas, posteriormente explicamos detalhadamente o modelo escolhido para a implementação, incluindo os detalhes de implementação utilizados, e finalmente apresentamos os resultados obtidos.

O processo de implementação foi complexo e difícil. Apesar de o modelo ser muito bem descrito e apresentar detalhes de implementação, ele possui algumas particularidades intrínsecas a seu funcionamento que foram difíceis de serem compreendidos a início. A abordagem do modelo, que utiliza diversos campos para a construção de um campo de potencial dinâmico, é uma abordagem de certa forma incomum e que nós nunca tínhamos tido contato, tornando seu entendimento mais complexo. Além disso, realizar *debugging* nesse tipo de aplicação é uma tarefa altamente complexa, uma vez que em uma grade de centenas a milhares de células construída iterativamente os erros se propagam muito rapidamente e são difíceis de serem rastreados e eliminados. Muitas vezes nós tivemos de imprimir os valores das células em diversas matrizes, para então poder adivinhar sua precedência e finalmente eliminá-los.

Apesar das dificuldades, nós implementamos com sucesso um modelo de animação comportamental com suporte a milhares de pedestres caminhando simultaneamente em tempo real. Nossa implementação, porém, não abordou todos os aspectos da simulação de multidões. O problema de renderização e gerenciamento de variedades não foi abordado por nós. Além disso, o modelo escolhido possui uma abordagem macroscópica da simulação de multidões, só simulando a movimentação, ignorando as possíveis interações que as pessoas podem realizar no dia a dia.

Como trabalhos futuros, nós propomos a implementação de um visualizador 3D capaz de lidar com as milhares de pessoas que fazem parte da simulação, possibilitando a renderização fiel em tempo real, com iluminação e sombra, e gerenciando as suas diferenças (cada pessoa deve parecer única). Outra proposta é a integração de agentes autônomos ao modelo: a implementação atual lidaria com a movimentação dos agentes, porém as pessoas apresentariam comportamento mais elaborado, possibilitando interações mais complexas do que somente movimentação em fluxo. Outra possibilidade é a implementação do modelo em processadores paralelos como GPUs (*Graphics Processing Units*): o atual algoritmo utilizado para a construção do campo potencial dinâmico, FMM, é serial, não podendo ser paralelizado; existem outras abordagens paralelas as quais podem ser investigadas, tais como o *Fast Interactive Method* [18].

## 6 Referencias

- [1] Aubel A., Boulic R., Thalmann D.: Real-time display of virtual humans: Levels of detail and impostors. *IEEE Transactions on Circuits and Systems for Video Technology* 10, 2 (2000), 207–217.
- [2] Bonabeau E., Dorigo M., Theraulaz G.: Inspiration for optimization from social insect behaviour. *Nature* 406 (2000), 39–42.
- [3] Bonabeau E., Dorigo M., Theraulaz G.: *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.
- [4] Brogan D., Hodgins J.: Group behaviors for systems with significant dynamics. *Autonomous Robots* 4 (1997), 137–153.
- [5] Bayazit O. B., Lien J.-M., Amato N. M.: Better group behaviors in complex environments using global roadmaps. In *Proc. Artificial Life '02* (2002).
- [6] Buckmann, L. T. and Leather, J. (1994). Modelling Station Congestion the PEDROUTE Way. *Traffic Engineering and Control*. 35(6): 373–377.
- [7] Couzin I. D., Krause J., Franks N. R., Levin S. A.: Effective leadership and decision-making in animal groups on the move. *Nature* 433 (2005), 513–516.
- [8] DeLoura M. (Ed.): *Game Programming Gems*. Charles River Media, 2000 p. 236.
- [9] Dobbyn S., Hamill J., O'Connor K., O'Sullivan C.: Geopostors: A real-time geometry/impostor crowd rendering system. In *Proceedings of ACM SIGGRAPH* (2005), ACM Press, p. 933.
- [10] Dorigo M.: *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italy, 1992.
- [11] Doyle A.: The two towers. *Computer Graphics World* (February 2003).
- [12] Farenc N., Boulic R., Thalmann D.: An informed environment dedicated to the simulation of virtual humans in urban context. In *Eurographics'99* (Milano, Italy, 1999), Brunet P., Scopigno R. (Eds.), Vol. 18, pp. 309–318.
- [13] Galea, E. R., Perez Galparsoro, J. M. and Pearce, J. (1993). A Brief Description of the EXODUS Evacuation Model. In. *Proc. International Conference on Fire Safety*, San Francisco, USA: 149–162.
- [14] Gilbert N.: *Simulation: An emergent perspective*. In *New Technologies in the Social Sciences* (Bournemouth, UK, 1995).
- [15] Hodgins J., Brogan D.: Robot herds: Group behaviors for systems with significant dynamics. In *Proc. Artificial Life IV* (1994), pp. 319–324.
- [16] Hamill J., McDonnel R., Dobbyn S., O'Sullivan C.: Perceptual evaluation of impostor representations for virtual humans and buildings. *Eurographics'05: Computer Graphics Forum* 24, 3 (September 2005), 581–590.

- [17] Jager W., Popping R., van de Sande H.: Clustering and fighting in two-party crowds: Simulating the approach–avoidance conflict. *Journal of Artificial Societies and Social Simulation* 4, 3 (2001).
- [18] Jeong, W. and Whitaker, R. T. 2008. A Fast Iterative Method for Eikonal Equations. *SIAM J. Sci. Comput.* 30, 5 (Jun. 2008), 2512–2534.
- [19] Kallmann M., Bieri H., Thalmann D.: Fully dynamic constrained Delaunay triangulations. In *Geometric Modelling for Scientific Visualization*, Brunnett G., Hamann B., Mueller H., Linsen L. (Eds.), Springer-Verlag, 2003, pp. 241–257.
- [20] Koeppel D.: Massive attack. *Popular Science* (November 2002).
- [21] Kuligowski, E. D. and Peacock, R. D. (2005). A Review of Building Evacuation Models, Fire Research Division. Building and Fire Research Laboratory. National Institute of Standards and Technology.
- [22] Lamarche F., Donikian S.: Crowds of virtual humans: A new approach for real time navigation in complex and structured environments. *Eurographics'04: Computer Graphics Forum* 23, 3 (September 2004), 509–518.
- [23] LeBon G.: *Psychologie des Foules*. Alcan, 1895.
- [24] McPhail C., Powers W., Tucker C.: Simulating individual and collective actions in temporary gatherings. *Social Science Computer Review* 10, 1 (Spring 1992), 1–28.
- [25] Musse S. R., Thalmann D.: A hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics* 7, 2 (April-June 2001), 152–164.
- [26] Musse S. R.: *Human Crowd Modelling with Various Levels of Behaviour Control*. PhD thesis, EPFL, Lausanne, 2000.
- [27] Niederberger C., Gross M.: Hierarchical and heterogeneous reactive agents for real-time applications. *Computer Graphics Forum* 22, 3 (2003) (Proc. Eurographics'03).
- [28] O'Sullivan C., Cassell J., Vilhj'almsson H., Dingliana J., Dobbyn S., McNamee B., Peters C., Giang T.: Levels of detail for crowds and groups. *Computer Graphics Forum* 21, 4 (Nov. 2002), 733–741.
- [29] Okazaki S., Matsushita S.: A study of simulation model for pedestrian movement with evacuation and queuing. In *Proc. International Conference on Engineering for Crowd Safety '93* (1993).
- [30] Pettr'e J., Laumond J.-P., Thalmann D.: A navigation graph for real-time crowd animation on multilayered and uneven terrain. In *First International Workshop on Crowd Simulation (V-CROWDS'05)* (2005), pp. 81–89.
- [31] Reynolds C. W.: Interaction with groups of autonomous characters. In *Proc. Game Developers Conference '00* (2000), pp. 449–460.

- [32] Reynolds C. W.: Flocks, herds, and schools: A distributed behavioral model. In Computer Graphics (ACM SIGGRAPH '87 Conference Proceedings) (Anaheim, CA, USA, 1987), Vol. 21, ACM, pp. 25–34.
- [33] Reynolds C. W.: Steering behaviors for autonomous characters. In Proceedings of Game Developers Conference 1999 (1999), pp. 763–782.
- [34] Reynolds, C. (2006). Big Fast Crowds on PS3. In. Proc. Sandbox (ACM SIGGRAPH symposium on Videogames), Boston, USA: 113–121.
- [35] Schweingruber D.: A computer simulation of a sociological experiment. Social Science Computer Review 13, 3 (1995), 351–359.
- [36] Scott R.: Sparking life: Notes on the performance capture sessions for 'The Lord of the Rings: The Two Towers.' ACM SIGGRAPH Computer Graphics 37, 4 (2003), 17–21.
- [37] Sung M., Gleicher M., Chenney S.: Scalable behaviors for crowd simulation. Computer Graphics Forum 23, 3 (2004). Proc. Eurographics '04.
- [38] Still G.: Crowd Dynamics. PhD thesis, Warwick University, 2000.
- [39] Treuille A., Cooper S., Popovic Z.: Continuum crowds. ACM Transactions on Graphics 25, 3 (2006), 1160–1168.
- [40] Thomas G., Donikian S.: Modeling virtual cities dedicated to behavioural animation. In Eurographics'00 (Interlaken, Switzerland, 2000), Gross M., Hopgood F. (Eds.), Vol. 19, pp. C71–C79.
- [41] Teknomo, G. (2002). Microscopic Pedestrian Flow Characteristics: Development of an Image Processing Data Collection and Simulation Model. Department of Human Social Information Sciences, Graduate School of Information Sciences, Tohoku University.
- [42] Tecchia F., Loscos C., Chrysanthou Y.: Image-based crowd rendering. IEEE Computer Graphics and Applications 22, 2 (March–April 2002), 36–43.
- [43] Tecchia F., Loscos C., Chrysanthou Y.: Visualizing crowds in real-time. Computer Graphics Forum 21, 4 (December 2002), 753–765.
- [44] Thompson P., Marchant E.: Testing and application of the computer model 'simulex.' Fire Safety Journal 24, 2 (1995), 149–166.
- [45] Tsitsiklis, J. N. 1995. Efficient algorithms for globally optimal trajectories. IEEE Transactions on Automatic Control 40, 9 (Sept.), 1528–1538.
- [46] Tucker C., Schweingruber D., McPhail C.: Simulating arcs and rings in temporary gatherings. International Journal of Human-Computer Systems 50 (1999), 581–588.
- [47] Tu X., Terzopoulos D.: Artificial fishes: Physics, locomotion, perception, behavior. In Computer Graphics (ACM SIGGRAPH '94 Conference Proceedings) (Orlando, FL, USA, 1994), Vol. 28, ACM, pp. 43–50.

- [48] Ulicny B., Thalmann D.: Crowd simulation for interactive virtual environments and VR training systems. In Proc. Eurographics Workshop on Animation and Simulation (2001), Springer-Verlag, pp. 163–170.
- [49] Ulicny B., Thalmann D.: Towards interactive real-time crowd behavior simulation. Computer Graphics Forum 21, 4 (Dec. 2002), 767–775.
- [50] Vaughan R. T., Sumpter N., Henderson J., Frost A., Cameron S.: Experiments in automatic flock control. Robotics and Autonomous Systems 31 (2000), 109–177.
- [51] Varner D., Scott D., Micheletti J., Aicella G.: UMSC Small Unit Leader Non-Lethal Trainer. In Proc. ITEC'98 (1998).
- [52] Williams J.: A Simulation Environment to Support Training for Large Scale Command and Control Tasks. PhD thesis, University of Leeds, 1995.
- [53] Wand M., Strasser W.: Multi-resolution rendering of complex animated scenes. Computer Graphics Forum 21, 3 (2002) (Proc. Eurographics'02).
- [54] Thalmann, D. and Musse, S. R. 2007 Crowd Simulation. Springer-Verlag New York, Inc.
- [55] Badler, N. 2008 Virtual Crowds: Methods, Simulation, and Control (Synthesis Lectures on Computer Graphics and Animation). Morgan and Claypool Publishers.