



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

RICARDO BUSTAMANTE DE QUEIROZ

**TÉCNICAS DE VISÃO PANORÂMICA E DE PERCEPÇÃO DE DENSIDADE PARA
MODELOS DE SIMULAÇÃO DE MULTIDÃO BASEADOS EM VISÃO SINTÉTICA**

FORTALEZA

2016

RICARDO BUSTAMANTE DE QUEIROZ

TÉCNICAS DE VISÃO PANORÂMICA E DE PERCEPÇÃO DE DENSIDADE PARA
MODELOS DE SIMULAÇÃO DE MULTIDÃO BASEADOS EM VISÃO SINTÉTICA

Dissertação apresentada ao Curso de do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação.

Orientador: Prof. Dr. Creto Augusto Vidal.

Coorientador: Prof. Dr. Joaquim Bento Cavalcante Neto.

FORTALEZA

2016

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

Q47t Queiroz, Ricardo Bustamante de.
Técnicas de Visão Panorâmica e de Percepção de Densidade para Modelos de Simulação de Multidão Baseados em Visão Sintética / Ricardo Bustamante de Queiroz. – 2016.
79 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2016.

Orientação: Prof. Dr. Creto Augusto Vidal.

Coorientação: Prof. Dr. Joaquim Bento Cavalcante Neto.

1. Computação Gráfica. 2. Realidade Virtual. 3. Simulação de Multidões. 4. Visão Sintética. I. Título.

RICARDO BUSTAMANTE DE QUEIROZ

TÉCNICAS DE VISÃO PANORÂMICA E DE PERCEPÇÃO DE DENSIDADE PARA
MODELOS DE SIMULAÇÃO DE MULTIDÃO BASEADOS EM VISÃO SINTÉTICA

Dissertação apresentada ao Curso de do
Programa de Pós-Graduação em Ciência da
Computação do Centro de Ciências da Universi-
dade Federal do Ceará, como requisito parcial
à obtenção do título de mestre em Ciência da
Computação. Área de Concentração: Ciência da
Computação.

Aprovada em: 28 / 03 / 2016.

BANCA EXAMINADORA

Prof. Dr. Creto Augusto Vidal (Orientador)
Universidade Federal do Ceará (UFC)

Joaquim Bento Cavalcante Neto (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Esdras Soares de Medeiros Filho
Universidade Federal do Ceará (UFC)

À minha mãe Ivina Marley Sá Bustamante, que, como professora, sempre me deu apoio e condições favoráveis para que eu pudesse estudar e seguir a minha carreira. Fico contente que ela possa ter acompanhado até o final do meu Mestrado, e sentido orgulho daquilo que ela ajudou a construir. Ela nos deixou alguns meses depois, mas seus ensinamentos me acompanharão para sempre.

RESUMO

Simular multidões é algo muito importante em uma grande variedade de aplicações de realidade virtual, pois a existência de avatares capazes de se comportar como uma multidão real melhora o senso de imersão dos usuários. Recentemente, modelos para simular multidões com personagens que possuem percepção visual, por serem equipados com um sistema de sensor visual sintético, têm trazido resultados interessantes em termos da naturalidade em que os personagens se locomovem em seu ambiente. Nesse trabalho, são propostas duas novas funcionalidades ao sistema visual dos personagens: a primeira é a substituição da projeção planar, originalmente proposta nos modelos para a visão dos agentes, por uma técnica de renderização de cenas panorâmicas, baseada em fatias, para permitir ângulos de abertura do campo de visão maiores que 180° . São avaliados os impactos da inclusão da técnica tanto em termos do desempenho do algoritmo como no comportamento do agente. A segunda é a adição da percepção de densidade de multidões no modelo de direção baseado em gradiente. Um estudo comparativo é feito sobre esse modelo adaptado com outra técnica existente.

Palavras-chave: Simulação de multidão. Visão sintética. Prevenção de colisão.

ABSTRACT

Crowd simulation is important in a wide variety of virtual reality applications, since the existence of avatars capable of behaving like a real crowd improves the user's immersion sense. Recently, models to simulate crowds with characters that have visual perception, for being equipped with a synthetic vision system, have brought interesting results in terms of naturality in which the characters move themselves through the environment. In this work, two new functionalities for the virtual characters synthetic vision systems are proposed: the first is the substitution of the planar projection, originally used for the vision of the agents, for a technique that renders panoramic scenes using slices, to allow field of views with angles greater than 180° . The impact of this change is evaluated both on the performance of the implementation, but also how it changes the behavior of the agent. The second addition is the perception of density of crowds over the gradient-based model. A comparative study is performed over this model with the existing technique.

Keywords: Crowd simulation. Synthetic vision. Collision avoidance.

LISTA DE FIGURAS

Figura 1 – Exemplos de multidões virtuais em um filme (superior), em um jogo (inferior esquerdo) e em um cenário de evacuação (inferior direito).	16
Figura 2 – Exemplo de técnicas de planejamento de trajetória: local (esquerda) e global (direita).	18
Figura 3 – Simulação de um modelo baseado em visão sintética (ONDŘEJ <i>et al.</i> , 2010). Padrões de comportamento surgem naturalmente de forma semelhante a multidões reais.	19
Figura 4 – Imagens geradas do ponto de vista do agente vermelho (acima) em um ambiente povoado por outros agentes (abaixo). Diversas informações sobre a dinâmica do ambiente e dos demais agentes são capturadas a partir do fluxo óptico do agente observador. O mesmo processo é repetido para todos os agentes na cena.	20
Figura 5 – Exemplo de modelo fazendo uso de regras. Buscar e fugir (esquerda). Perseguir e esquivar (direita).	26
Figura 6 – Efeito de gargalo reproduzido por Helbing <i>et al.</i> (2000).	26
Figura 7 – Multidão sendo repelida por fumaça enquanto tentam evacuar a construção.	27
Figura 8 – Simulação de evacuação de uma sala usando modelo de autômato celular. É possível notar a organização dos agentes na grade regular.	28
Figura 9 – Exemplos de campos de orientação. Campo atualizado por interação com usuário (esquerda) e campo egocêntrico (direita).	29
Figura 10 – Exemplo de campo multipotencial. Oito campos potenciais são usados para mudar momentaneamente o objetivo do agente.	30
Figura 11 – (a) Uma configuração com oito agentes, onde suas velocidades são indicadas pelas setas. (b) Os semi-planos indicam as possíveis velocidades para o agente A, com respeito aos demais agentes. A seta indica a velocidade atual de A.	31
Figura 12 – Dados reais (esquerda) e comportamento reproduzido (direita).	32
Figura 13 – Dados reais (cima) e comportamento reproduzido (baixo).	33
Figura 14 – Visão como uma área geométrica. Visão de um peixe (esquerda) e visão de um boid (direita).	34

Figura 15 – Visão de um agente no modelo proposto por Ondřej <i>et al.</i> (2010). As informações sobre a dinâmica dos obstáculos ($\dot{\alpha}$ e t_{ti}) são passadas para o agente através de seu sensor visual. A cor vermelha foi utilizada para representar valores pequenos.	34
Figura 16 – O ângulo de abertura α e sua derivada no tempo $\dot{\alpha}$ permitem a detecção de colisões futuras. Da perspectiva de um observador (personagem de baixo), uma colisão é prevista quando α não varia ao longo do tempo, ou seja, o valor de $\dot{\alpha}$ é 0 (centro). Caso o ângulo esteja variando ao longo do tempo, um dos dois agentes irá cruzar o ponto de colisão antes do outro, evitando o contato (esquerda, direita).	37
Figura 17 – A função de limiar definida por Ondřej <i>et al.</i> (2010). Há risco de colisão quando o par $(\dot{\alpha}, t_{ti})$ está abaixo da linha da função, e com valores $t_{ti} > 0$. Também é possível ver que, quanto menor o valor de t_{ti} , maior será a reação do agente.	38
Figura 18 – Comparação entre três modelos feitas em Hughes <i>et al.</i> (2015), em uma cena circular com 100 agentes. É possível notar que a aglomeração no centro da cena foi evitada graças à percepção de um aumento de densidade na multidão. Comparando com dados de captura, foi observado que pessoas tentam evitar situações de congestionamento.	39

- Figura 19 – As figuras ilustram quatro situações diferentes com dois agentes. A linha superior mostra as posições dos agentes e suas velocidades, e a linha inferior, a distância entre os agentes ao longo do tempo. Na linha inferior, o ponto verde no gráfico indica o par $(dca, ttca)$ para cada uma das situações. (a) e (b) representam situações onde os agentes estão convergindo, em (c) os agentes se movem paralelamente e em (d), as trajetórias dos agentes divergem. Em (a), o valor de dca é 0 com $ttca$ em 1, o que indica que em 1 segundo acontecerá uma colisão entre os dois agentes. Em (b), o valor de dca indica que não há um risco de colisão, pois seu valor é maior que a soma dos raios dos dois agentes. Em (c) o valor de dca é constante e o valor de $ttca$ é indeterminado, pois os agentes não estão nem se aproximando e nem se afastando, mantendo uma distância constante. Em (d), o valor de $ttca$ é negativo, indicando que os pontos em que as trajetórias dos agentes se encontram estão na direção oposta da qual pelo menos um deles está indo. 41
- Figura 20 – (esquerda) Função de custo de movimento C_m , definida como a soma de duas funções Gaussianas relativas ao ângulo do agente em relação ao seu objetivo e sua velocidade atual em relação a sua velocidade de conforto. (direita) Plot da função de custo C_o em relação aos valores de dca e $ttca$ 42
- Figura 21 – Exemplos de campos de visão com *frustum* de projeção sendo observados por cima da cabeça dos avatares. A imagem da esquerda mostra o exemplo com um ângulo menor que 180° . Na imagem do meio, um exemplo com ângulo de 180° . Nesse caso os planos *near* e *far* coincidem, ou seja, o frustum não possui volume o que faz com que nada seja projetado no plano e conseqüentemente a cena não seja renderizada. A imagem da direita mostra um exemplo com um ângulo maior que 180° . Nesse caso, o problema é que o plano *far* fica atrás do plano *near*. 44
- Figura 22 – Visão de cima, comparando diferentes ângulos de visão horizontal (120° , 180° , 240°), com 1 e 5 fatias, respectivamente. 46
- Figura 23 – Na imagem, a visão do agente em destaque ao centro é reproduzida para diferentes campos de visão no cenário circular exibido à esquerda. As imagens da direita representam a visão do agente para um ângulo horizontal variando de 90° a 360° , com um ângulo vertical fixo de 80° 46

Figura 24 – Na primeira imagem vê-se um panorama real. Na segunda imagem, um panorama renderizado utilizando a técnica, com ângulo de 270°. Na terceira imagem, a cena renderizada na segunda imagem vista por cima do agente.	48
Figura 25 – Menor ângulo necessário para que dois agentes possam se ver. A situação ilustrada é o caso extremo em que dois agentes estão colados lado a lado e qualquer alteração de rotação que possa causar uma colisão fará com que os agentes se vejam antes. 2α representa o valor total de abertura do campo de visão.	49
Figura 26 – Ilustração mostrando como é calculado o menor ângulo com as considerações da Figura 25. Sabendo que o agente possui raio R , e ao estar lado a lado com outro agente a distância entre seus centros é de $2R$, podemos calcular o ângulo mínimo de visão 2α . O valor de α é $\arccos(\frac{R}{2R}) = 60^\circ$. Portanto o ângulo de visão deve ser no mínimo 120°	49
Figura 27 – Alguns testes realizados com o campo de visão variando o ângulo horizontal. Em (a), a configuração do cenário. O agente vermelho e o agente azul caminham lado a lado, paralelamente, enquanto o agente verde, mais rápido, tenta uma ultrapassagem. Posição inicial dos agentes e seus objetivos, marcados por quadrados de suas respectivas cores. Em (b,c,d) é utilizado o modelo OSV, e em (e,f,g) é utilizado o modelo GBM. Em (b,e) o ângulo de 30° é utilizado. Em (c,f) o valor é de 150° . Para (d,g) o valor é de 270°	50
Figura 28 – Estudo do ângulo vertical e da orientação da visão do agente. Na esquerda, são testados nos dois modelos um ângulo de visão reduzido, de 40° de abertura e -60° de orientação. Na direita, são testados os valores padrão dos modelos, com abertura de 80° e orientação de -40°	51
Figura 29 – Estudo do impacto do número de fatias no comportamento do agente. Mesmo a imagem resultante da renderização sendo diferente, pouca variação foi observada na maioria dos testes. Em (a), a configuração inicial, semelhante à Figura 27. Em (b,c,d) é testado o modelo OSV, enquanto em (e,f,g) é testado o modelo GBM. Em (b,e) os modelos são testados com sua visão padrão, com apenas uma fatia. Em (b,e) é testado com 4 fatias, e em (c,f) com 16 fatias.	52

- Figura 30 – O resultado do teste de desempenho. O primeiro gráfico representa a *FPS* para o número de fatias. O segundo gráfico a influência do ângulo horizontal na *FPS*. E na terceira imagem, é analisado o número de fatias na renderização. Em azul claro encontram-se os resultados obtidos com o modelo OSV, e em azul escuro, os resultados obtidos com o modelo GBM. 53
- Figura 31 – Diferentes cenários usados nos testes. Na primeira linha, uma cena circular, onde os agentes encontram-se equidistantes do centro, e tem como objetivo o ponto diametralmente oposto. Na segunda linha, o caso onde dois grupos devem se atravessar para atingir seus objetivos, localizados na posição em que os agentes do outro grupo se encontram. Na terceira linha, dois grupos marcham em direções que são perpendiculares entre si. Da esquerda para a direita, as figuras indicam a passagem do tempo. Na primeira está a configuração inicial da cena, e na segunda está um momento em que há um grande número de agentes desviando dos demais para conseguirem atingir seu objetivo, e na terceira, o momento que os agentes chegam ao seu destino. 55
- Figura 32 – Comparando diferentes espécies de agentes e seus respectivos campos de visão. Do lado esquerdo, um ser humano sendo visto por cima. Do lado direito, um peixinho dourado. A região azul corresponde ao campo de visão do olho esquerdo. A região vermelha a do olho direito. A região de cor roxa indica onde há sobreposição das imagens dos dois olhos, composta no cérebro do indivíduo. A região roxa é denominada de área de visão binocular, ou estereoscópica, e as demais regiões são áreas de visão monocular, ou visão periférica. Os ângulos indicados na imagem não correspondem às medidas reais, sendo apenas ilustrativos das diferentes regiões. 56
- Figura 33 – O agente tolera ângulos em relação ao seu objetivo. À medida que se aproxima de seu objetivo, o agente deveria tolerar menos desvios de sua trajetória. Nesse caso, o agente tem que manter a direção do objetivo dentro do ângulo de tolerância a sua frente. Na imagem, o agente com campo de tolerância verde permite uma variação menor do que o agente que possui o campo vermelho por estar mais próximo do objetivo. 58

Figura 34 – Como é feito o cálculo da densidade. O agente azul é o observador, e os demais são obstáculos. Na parte superior, é mostrada a cena vista do alto, sobre a cabeça dos agentes. Na parte inferior, é mostrada uma visão lateral da mesma cena. Na esquerda é ilustrada a configuração da cena em um determinado estado, e na direita, o gráfico da função que indica a contribuição para o fator de densidade de cada obstáculo é sobreposto. O gráfico descreve uma função gaussiana cujo parâmetro é apenas a distância, onde a contribuição para o fator da densidade é maior ao longo do eixo da velocidade do agente do que em relação ao eixo perpendicular, o que significa que obstáculos aglomerados na frente do agente representam um risco maior de colisão. Visto de cima, a equação para cada posição parece elíptica por causa da prioridade que se dá para o afastamento no eixo perpendicular ao movimento do agente. 60

Figura 35 – Cálculo do fator de densidade. Na figura (a), vemos o estado atual da cena. Nosso cálculo será referente à visão do agente azul. Em (b), extrapolamos a posição em que os personagens estarão após um *timestep*, deslocando os agentes na direção de suas velocidades. Na figura (c), um deslocamento proporcional à distância de cada agente para o eixo da velocidade do observador é aplicado, para diminuir a relevância para o cálculo da densidade de objetos que não estejam na trajetória do personagem azul. Em (d), vemos as posições finais após todas as transformações. Essas distâncias entre os agentes vermelhos e o agente azul serão utilizadas para o cálculo do fator de densidade. 62

Figura 36 – Um exemplo de testes comparativos, usando os valores $\sigma_1 = 1$ e $\sigma_2 = 0.2$. O cenário circular com 8 agentes foi utilizado. Nas imagens superiores, o cenário de círculo com raio pequeno foi utilizado, enquanto nas imagens inferiores, um círculo com maior raio foi utilizado. Na esquerda, encontra-se o modelo GBM, enquanto na direita está o GBM-D. É possível ver que, em situações de alta densidade, para os parâmetros utilizados, houve uma dispersão muito grande entre os agentes, levando a um comportamento indesejado. Porém, no caso do círculo maior, em que há maior espaço para os agentes percorrerem, os agentes se mantiveram mais espalhados, como previsto. . . 63

Figura 37 – Outro exemplo, usando a cena onde dois grupos devem trocar de lugar. A estrutura segue a mesma ordem da Figura 36	64
Figura 38 – Caso o parâmetro σ_1 seja muito alto, a função de custo da densidade irá predominar sobre as demais funções de custo, e os agentes jamais irão evitar colisões ou ir em direção ao seu objetivo, pois quanto mais se afastarem, menor será a densidade. O experimento exagerado mostra que a função de custo de fato funciona, mesmo que criando um resultado indesejado. É possível ver um padrão de zigue zague na imagem, pois quando não consegue ver nenhum outro agente, ele começa a seguir a função de custo do objetivo C_o , mas retoma seu afastamento dos demais quando algum agente entra em seu campo de visão.	65
Figura 39 – Ilustrando o resultado do algoritmo com 3 fatias e um ângulo total de 180° . Três frustums vão ser criados, com ângulo horizontal de 60° . O Valor indicado na imagem para cada frustum (0, 1 e 2) corresponde ao ângulo que a projeção é rotacionada em torno do eixo Y.	75

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Contextualização	17
1.2	Objetivos	19
1.3	Proposta	21
1.4	Organização	21
2	TRABALHOS RELACIONADOS	23
2.1	Técnicas de simulação de multidões	23
2.1.1	<i>Planejamento de trajetória global</i>	24
2.1.2	<i>Planejamento de trajetória local</i>	24
2.1.2.1	<i>Modelos baseados em regras</i>	25
2.1.2.2	<i>Modelos baseados em partículas</i>	25
2.1.2.3	<i>Modelos de autômato celular</i>	28
2.1.2.4	<i>Modelos de campo de orientação</i>	28
2.1.2.5	<i>Modelos baseados em velocidade</i>	30
2.1.2.6	<i>Modelos orientados por exemplos</i>	31
2.1.2.7	<i>Modelos baseados em visão sintética</i>	33
2.1.2.8	<i>Modelos híbridos</i>	35
2.2	Modelo proposto por Ondřej	36
2.3	Modelo adaptativo de densidade	39
2.4	Modelo Baseado em Gradiente	40
2.5	Considerações finais	42
3	VISÃO PANORÂMICA	44
3.1	Campo de visão em modelos baseados em visão sintética	44
3.2	Técnica Proposta	45
3.3	Implementação	45
3.4	Resultados	47
3.4.1	<i>Visão panorâmica</i>	48
3.4.2	<i>Comportamento dos Agentes</i>	49
3.4.3	<i>Desempenho</i>	53
3.5	Trabalhos futuros	54

3.5.1	<i>Diferentes regiões de visão</i>	54
3.5.2	<i>Diversificação dos agentes</i>	57
3.5.3	<i>Relaxamento da restrição do objetivo</i>	57
3.5.4	<i>Movimentação em três dimensões</i>	57
3.6	Considerações finais	58
4	DENSIDADE	59
4.1	Percepção de densidade	59
4.2	Técnica proposta	60
4.3	Implementação	61
4.4	Resultados	62
4.5	Trabalhos futuros	65
4.5.1	<i>Buscar novos cenários</i>	65
4.5.2	<i>Comparação com HSV</i>	66
4.6	Considerações finais	66
	REFERÊNCIAS	67
	APÊNDICE A – PROJEÇÃO PANORÂMICA	73
	APÊNDICE B – GRADIENTE DA FUNÇÃO DE CUSTO	76
	APÊNDICE C – DERIVADAS PARCIAIS DA FUNÇÃO DE CUSTO	77

1 INTRODUÇÃO

Uma multidão é "*qualquer agrupamento humano contendo um número grande de indivíduos atuando ao mesmo tempo, não necessariamente em comum, dentro de um ambiente*"¹. Uma multidão virtual, de maneira semelhante, é composta por entidades virtuais denominadas agentes. Cada um desses agentes possui um objetivo individual sendo este, por exemplo, chegar em um determinado local. Para cumprir seu objetivo, ele deve respeitar as leis da física e deve desviar dos obstáculos estáticos e dinâmicos compostos pelo cenário e por demais agentes da multidão.

Figura 1 – Exemplos de multidões virtuais em um filme (superior), em um jogo (inferior esquerdo) e em um cenário de evacuação (inferior direito).



Fonte: The Hobbit: The Battle of the Five Armies (Warner Bros. PicturesMo, 2014), Assassin's Creed Syndicate (Ubisoft, 2015), Pelechano *et al.* (2008).

O campo de estudo de simulação de multidões aborda a criação de multidões virtuais. Esses estudos são motivados pelas diversas aplicações de simulações de multidões, incluindo engenharia civil, projeto arquitetônico e indústria do entretenimento, que inclui a criação de filmes e jogos eletrônicos (Figura 1).

¹ <https://pt.wikipedia.org/wiki/Multitude>

Cada uma dessas aplicações demanda diferentes níveis de qualidade da simulação e de verossimilhança. Para filmes, uma animação (visual e comportamental) de alta qualidade é priorizada, podendo essa qualidade ser medida de várias formas diferentes, dependendo do tipo de agente ou se o filme se trata de algo real ou fantasioso. O desempenho computacional deve ser o melhor possível para reduzir o tempo de produção do filme, mas pode ser sacrificado em detrimento da qualidade final do produto, pois o resultado só necessita ser gerado uma vez e pode ser exibido diversas vezes, sem custo computacional adicional. Em jogos e outras aplicações de realidade virtual, é preciso encontrar um equilíbrio entre qualidade da animação e desempenho, já que a simulação precisa ser executada em tempo real para garantir a interatividade com o usuário. Para engenharia civil, projetos arquitetônicos e simulações em geral, a prioridade está em produzir o resultado mais fiel, tanto em termos de comportamento dos agentes, como em relação à situação em que se deseja simular, enquanto a qualidade da animação e o desempenho são fatores secundários.

Mesmo com todos esses avanços na área, ainda existe um longo caminho para conseguir compreender todos os fatores que afetam o comportamento dos personagens virtuais, como, por exemplo, o modo de locomoção e percepção dos seres humanos. É conhecido pela literatura que o fluxo ótico tem bastante importância na locomoção ((GIBSON, 1958);(CUTTING *et al.*, 1995)). Dos sentidos humanos, a simulação da visão é a mais direta de se imaginar a partir de uma aplicação de computação gráfica, pois seus conceitos foram baseados na forma que este sentido humano atua. Então, aprofundar o entendimento da visão pode contribuir com o objetivo de reproduzir sinteticamente o sentido e possibilitar simulações mais críveis.

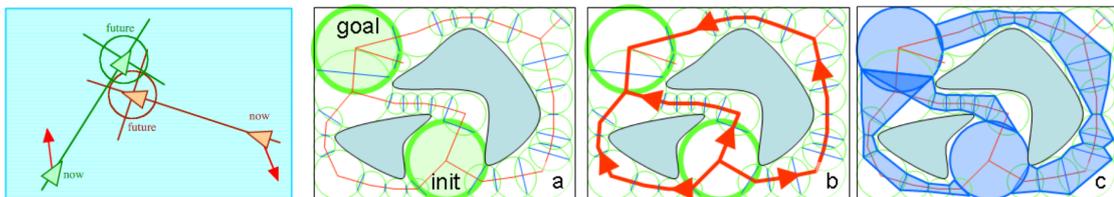
1.1 Contextualização

A área de estudos de simulação de multidões abrange diversas sub-áreas necessárias para que no fim possa resultar em comportamentos observados em multidões reais. Uma delas tenta resolver o problema de *planejamento de trajetória*. É uma tarefa fundamental para conseguir simular multidões, e conseqüentemente, uma das que recebem maior atenção da comunidade de pesquisadores.

Planejamento de trajetórias por si só é uma área bem extensa e engloba diversos aspectos da locomoção dos personagens, e por conta disso pode ser categorizada de duas formas diferentes: *Planejamento local* e *planejamento global* (Figura 2). Técnicas de planejamento local tentam resolver o problema de colisão nas imediações do agente virtual. As técnicas de

planejamento global levam em consideração a trajetória completa do agente, e como ele deve navegar em um ambiente complexo. As duas categorias devem ser usadas em conjunto para possibilitar a navegação do agente entre diversos pontos chaves na cena, fazendo uso de uma técnica global, e entre cada dois pontos chaves, e evitar colisões com o cenário e outros agentes, fazendo uso de planejadores locais.

Figura 2 – Exemplo de técnicas de planejamento de trajetória: local (esquerda) e global (direita).



Fonte: Reynolds (1999), Pettré *et al.* (2005).

Uma classe de algoritmos baseados em agente de planejamento local, também denominados algoritmos *baseados em velocidade*, permite simuladores de multidão adquirirem um nível maior de realismo. Esses algoritmos usam informação de velocidade dos agentes e dos obstáculos na cena para calcular o risco de colisão em um futuro momento, e permitir que os agentes reajam de maneira antecipada a esse risco, imitando o comportamento inteligente do ser humano.

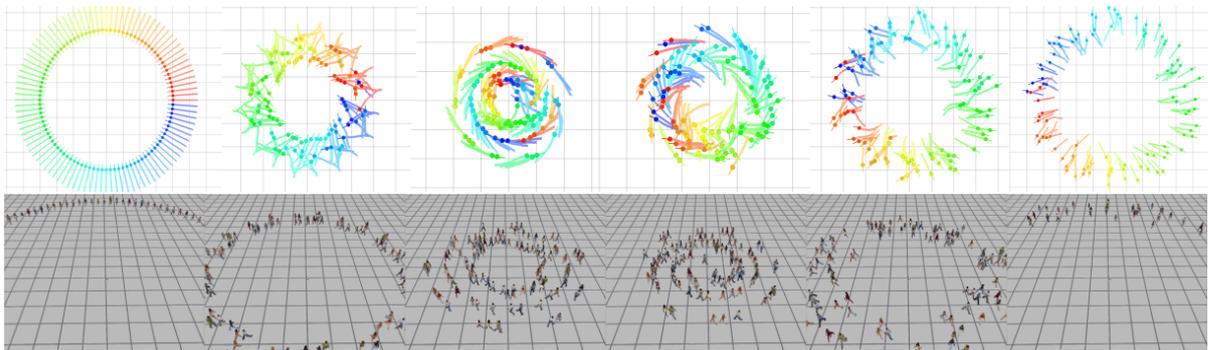
Uma categoria desses algoritmos baseados em velocidade capta essas informações através de sensores visuais equipados em agentes. Esses algoritmos, *baseados em visão sintética*, processam a informação visual a fim de obter informações sobre a dinâmica dos obstáculos, e utilizam essa informação para alimentar os controladores dos agentes que irão guiá-los pelo ambiente dinâmico.

Esses algoritmos têm sido vistos com ceticismo dado seu alto custo computacional. Apesar disso, eles têm demonstrado vários avanços em termos de simulação de multidões visualmente agradáveis. O primeiro resultado obtido em uma simulação de multidões usando a técnica de visão sintética foi obtido por Ondřej *et al.* (2010), abrindo caminho para um novo ramo de técnicas de simulação de multidão.

No trabalho de Ondřej *et al.* (2010), o agente utiliza sua informação visual para descobrir se há alguma ameaça iminente de colisão, e estimar quanto tempo levará para acontecer. Baseado nessas informações, que são fornecidas como entrada para o *loop* de controle do agente, o mesmo é capaz de mudar sua trajetória girando em torno de seu eixo ou reduzindo sua

velocidade, de forma a evitar a ameaça detectada. Algumas limitações do algoritmo são o fato de que ele utiliza apenas a informação de uma ameaça para definir sua estratégia de esquiva, e que ele não avalia as consequências de sua manobra, o que pode, eventualmente, resultar em colisões com outros obstáculos próximos. O agente pode, por exemplo, colidir com um agente à sua esquerda ao tentar desviar de um agente que está imediatamente a sua direita.

Figura 3 – Simulação de um modelo baseado em visão sintética (ONDŘEJ *et al.*, 2010). Padrões de comportamento surgem naturalmente de forma semelhante a multidões reais.



Fonte: Ondřej *et al.* (2010).

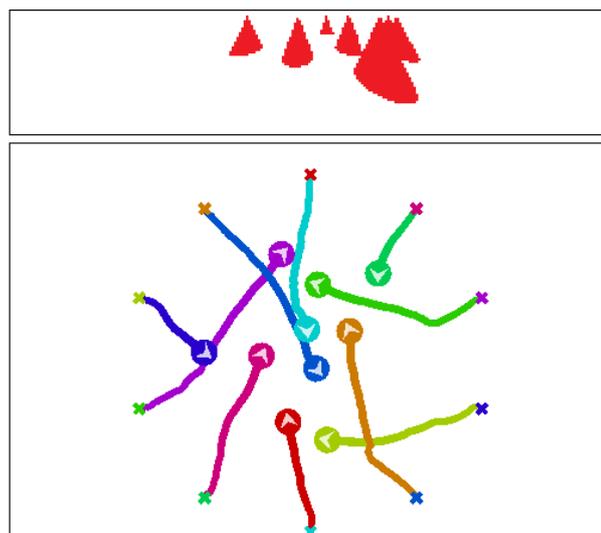
Dutra *et al.* (2015) fizeram avanços sobre o trabalho de Ondřej *et al.* (2010) mudando a forma que o agente utiliza as informações captadas pelo sensor visual. No lugar de considerar apenas uma ameaça para definir sua manobra, o agente passa a considerar tudo que está em seu campo de visão. Com essa informação, ele é capaz de avaliar o risco da situação como um todo, e é capaz de avaliar qual sua nova configuração de rotação e velocidade que reduz esse risco. Isso é feito através do cálculo de uma função de custo que leva em consideração o tempo e a distância para colisões iminentes, e fornecendo informações para o controlador do agente de qual orientação e velocidade ele deve assumir para reduzir as chances de colisões. Algumas limitações da técnica incluem o campo de visão reduzido, que por não ser capaz de representar a movimentação da cabeça e a visão periférica dos agentes, acaba permitindo que colisões aconteçam nessas regiões fora de cobertura; e que mesmo desviando dos riscos da melhor maneira possível, o agente não é capaz de prever situações de alta densidade, o que causa congestionamentos e possíveis travamentos dos agentes.

1.2 Objetivos

O objetivo deste trabalho é adicionar novas capacidades aos algoritmos baseados em visão sintética para expandir seu potencial e resolver problemas existentes nas técnicas

atuais. Estes algoritmos são capazes de direcionar os agentes usando o aparato visual de maneira semelhante à realizada pelos seres humanos. É feita uma abstração a respeito da natureza do movimento dos obstáculos, tratando obstáculos estáticos e dinâmicos da mesma maneira, considerando que objetos estáticos simplesmente possuem velocidade nula. Além disso, as formas reais dos objetos são levadas em consideração, uma vez que a cena é renderizada na retina de cada agente tal qual foi modelada. Como é usado um campo visual, a captura de dados causa um filtro natural de quais obstáculos são relevantes para a movimentação, seja pelo ângulo do campo de visão, ou por eventuais oclusões causadas por outros obstáculos ou agentes obstruindo a vista do observador. Por conta disso, é esperado um realismo maior em relação a outras técnicas que utilizam informações de uma natureza diferente, pois imita a forma que o ser humano captura as informações de seu ambiente. Estruturas de movimentação de grupos auto-organizadas emergem de forma natural, da mesma forma que são observadas no mundo real. Além disso, a pesquisa na área pode contribuir com outros campos de pesquisa, como a Neurociência, ajudando a compreender melhor como humanos se comportam em multidões e que informações são relevantes para sua movimentação.

Figura 4 – Imagens geradas do ponto de vista do agente vermelho (acima) em um ambiente povoado por outros agentes (abaixo). Diversas informações sobre a dinâmica do ambiente e dos demais agentes são capturadas a partir do fluxo óptico do agente observador. O mesmo processo é repetido para todos os agentes na cena.



Fonte: Autor.

A ideia desta dissertação é observar as técnicas existentes e propor algumas melhorias que resolvam determinados problemas relacionados a cada uma das diferentes simulações. Os objetivos deste trabalho são:

1. Estudar diferentes algoritmos baseados em visão sintética e avaliar o impacto que determinadas melhorias podem ter em seu resultado.
2. Propor uma técnica para melhorar o campo de visão de várias técnicas mudando apenas o passo de captura de dados, permitindo que os algoritmos continuem executando como antes.
3. Combinar a ideia de percepção de densidade, semelhante à que foi proposta em Hughes *et al.* (2015) para o modelo de Ondřej *et al.* (2010), aplicada ao modelo baseado em gradiente proposto por Dutra *et al.* (2015).

1.3 Proposta

Neste trabalho, primeiramente é proposta uma modificação nos algoritmos baseados em visão sintética que permite uma alteração no campo visual horizontal do agente para ângulos no intervalo de 0° a 360° . A ideia é substituir a etapa onde é calculado o campo de visão dos agentes virtuais, que geralmente é uma projeção planar em formato de tronco de pirâmide baseada na câmera *pinhole*, por uma projeção diferente que imita o comportamento de uma câmera panorâmica. O resultado dessa captura continua sendo uma imagem de mesmo tamanho que o algoritmo usava anteriormente, permitindo que essa técnica seja acoplada, sem muita adaptação, em todos os algoritmos baseados em visão sintética existentes.

Também é elaborada uma técnica para incluir no modelo de Dutra *et al.* (2015) o cálculo de densidade de multidões como é proposto por Best *et al.* (2014). A ideia é que o agente seja capaz de antecipar situações de aglomeração com antecedência, que podem levar a situações de travamento ou colisões. Uma adaptação inicial para inclusão dessa informação foi feita por Hughes *et al.* (2015), onde um valor de densidade é calculado em um ponto na frente de cada agente, e esse valor é utilizado na fórmula que define o nível da ameaça.

1.4 Organização

O restante desta dissertação está organizada da forma a seguir. No Capítulo 2, são listados e discutidos trabalhos relacionados relevantes. Neste capítulo, técnicas importantes são categorizadas e explicadas em detalhes. E por fim, para que se possa entender este trabalho e as melhorias propostas, o trabalho de Ondřej que simula multidões baseando-se em visão sintética será discutido, junto com o trabalho de Dutra, que propôs uma melhoria sobre este modelo

mudando a forma em que os dados capturados pela visão são utilizados para guiar o agente.

No Capítulo 3, a primeira melhoria proposta sobre os modelos de visão sintética é descrita. Nele, é explicada a deficiência observada no campo de visão dos agentes, e é proposta uma nova técnica para captura de imagens para os agentes virtuais.

No Capítulo 4, é feito um experimento em adicionar informação a respeito da densidade da multidão ao campo visual dos agentes que utilizam o modelo proposto por Dutra *et al.* (2015), de forma semelhante à que Hughes *et al.* (2015) propuseram, mas de forma a manter as informações sendo capturadas pelo campo visual do agente.

2 TRABALHOS RELACIONADOS

Este capítulo é dividido em duas partes. Na primeira parte são explicadas técnicas de planejamento global e de planejamento local, e em seguida as técnicas de planejamento local mais relevantes da literatura são apresentadas. Essas técnicas são categorizadas e separadas em diferentes seções. Na segunda parte, os modelos de visão sintética de Ondřej *et al.* (2010) e Dutra *et al.* (2015) são apresentados com mais detalhes. Por fim, serão discutidas duas propostas de melhoria para os modelos que permitem superar problemas existentes nos mesmos.

2.1 Técnicas de simulação de multidões

Um dos principais objetivos de simular multidões é computar a movimentação de vários personagens que resultam em comportamentos coletivos, o que ocasionou o surgimento de muitas soluções que podem ser encontradas na literatura, como revistas e livros recentes (PELECHANO *et al.*, 2008; THALMANN; MUSSE, 2013; ALI *et al.*, 2013)).

Animar multidões não envolve apenas o planejamento da trajetória dos avatares, mas envolve também diversos aspectos como: animação de avatares, caracterização, técnicas de renderização de multidões e desenvolvimento de interfaces de manipulação dessas multidões.

O foco deste trabalho é o planejamento de trajetória, ou planejamento de caminho. Como apresentado no capítulo anterior, existem planejadores locais e globais. Para ambientes simples, apenas reagir localmente a potenciais colisões pode ser suficiente para o agente esquivar de colisões e atingir seu objetivo. Isto é o trabalho de planejadores locais. Entretanto, quando o ambiente se torna mais complexo, planejadores locais podem levar os agentes a certas situações indesejadas. Isso pode acontecer por causa de situações contraditórias, onde, por exemplo, para chegar em seu objetivo o agente deve se afastar dele antes de retomar a rota de aproximação. Um planejador global define conjuntos de pontos de passagem pelo qual o agente deve passar até chegar em seu objetivo. A navegação entre dois desses pontos de passagem pode ser feita utilizando um planejador local.

Resumindo, um planejador global é responsável por planejar uma rota do ponto inicial até o ponto final, considerando as peculiaridades de cada ambiente, enquanto o planejador local está responsável por adaptar a rota entre dois pontos de forma que o agente possa desviar de obstáculos dinâmicos ou detalhes relativamente pequenos ignorados pelo planejador global. Para compreender melhor o planejamento de caminhos, inicialmente são abordados alguns

planejadores globais da literatura.

2.1.1 Planejamento de trajetória global

De acordo com Kapadia e Badler (2013): "Navegação em ambientes arbitrariamente grandes e complexos requer que um agente seja equipado com um modelo mental que provê alguma representação geométrica semanticamente significativa do mundo ao seu redor". O agente precisa de um modelo mental, ou um mapa do ambiente e precisa ser capaz de entendê-lo e usá-lo para atingir seu objetivo pelo melhor caminho possível. Este "mapa mental" pode ser criado a partir de um processamento do ambiente, onde os caminhos navegáveis serão estabelecidos. A maior parte dos planejadores globais consiste em computar o menor caminho entre vértices de um grafo. O que difere um do outro são detalhes relacionados a: formação do grafo; posicionamento dos vértices; as funções atribuídas aos vértices e arestas; e ao algoritmo usado para computar o menor caminho. Algumas categorias de abordagens baseadas na literatura (PELECHANO *et al.*, 2008; THALMANN; MUSSE, 2013; ALI *et al.*, 2013) podem ser citadas: planejamento de movimento discreto, roteiros probabilísticos, grafos de célula e portais, métodos reativos, campos de orientação e modelagem de ambiente.

2.1.2 Planejamento de trajetória local

Planejamento de caminho local, como definido por Kapadia e Badler (2013), é uma camada de inteligência que faz interface com a navegação ao longo de sua trajetória planejada, realizando sucessivas buscas locais levando em consideração as restrições de locomoção, sejam as limitações naturais do próprio agente, ou restrições impostas pelos obstáculos na cena.

Por ser a área que tem maior impacto na forma com que o agente se comporta em menor escala, o que afeta diretamente a qualidade da simulação, esta é uma das subáreas mais exploradas. As várias soluções para este problema podem ser classificadas de diferentes formas. Alguns autores optam por classificar de acordo com a forma em que os agentes são gerenciados, podendo ser abordagens macroscópicas ou microscópicas, sendo as microscópicas as que se preocupam em gerenciar o comportamento de cada agente individualmente, o que acaba por resultar em comportamentos globais emergindo naturalmente, e técnicas macroscópicas que já tentam definir tais comportamentos. Outros autores preferem classificar as técnicas de acordo com a capacidade de previsão de colisão. As abordagens são então classificadas como preditivas ou reativas, dependendo se elas permitem antecipação de suas reações, ou se o

movimento é inteiramente reativo às interações. Em (ZHENG *et al.*, 2009) os autores apresentam características para classificar técnicas de simulação de evacuação.

Neste trabalho, os algoritmos foram divididos de acordo com a técnica escolhida para evitar colisões. São estas: baseada em regras, baseada em partículas, automato celular, campos de orientação, baseados em velocidade, movidos a dados, baseados em visão sintética e híbridos.

2.1.2.1 Modelos baseados em regras

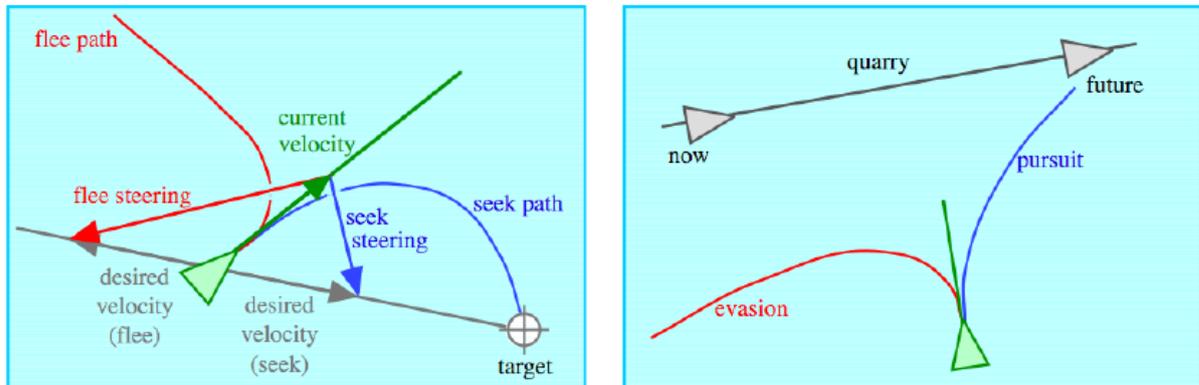
Reynolds (1987) apresentaram um modelo baseado em regras, onde o conceito de *boids* ("bird-oid") foi introduzida. Os autores fizeram esses agentes, denominados *boids*, se comportarem como um bando através da combinação de três regras individuais, onde os boids deveriam: Evitar colisões (separação), manter a mesma velocidade (alinhamento) e manter-se próximos aos demais (coesão). Modelos baseados em regra permitem caracterizar os agentes individualmente com comportamentos complexos formados pela composição de comportamentos simples. Reynolds (1999) expandiram seu trabalho anterior definindo comportamentos de direção para agentes autônomos, como exemplo: buscar, fugir, perseguir e esquivar.

Modelos baseados em regras têm a vantagem de permitir a modelagem de agentes heterogêneos. Entretanto, em alguns casos, pode ser difícil combinar regras, levando ao surgimento de artefatos no comportamento do agente, como por exemplo, regras contraditórias. Também pode ser computacionalmente caro gerenciar multidões com diversos tipos de comportamentos diferentes. Para atingir resultados com melhor performance, permitindo simulações em tempo real com diversos agentes, algumas técnicas foram propostas como: posicionamento espacial (REYNOLDS, 2000; REYNOLDS, 2006), agrupamento hierárquico (MUSSE, 2000; MUSSE; THALMANN, 2001) ou o uso de estruturas hierárquicas (LAMARCHE; DONIKIAN, 2004; SHAO; TERZOPOULOS, 2007).

2.1.2.2 Modelos baseados em partículas

Pesquisadores perceberam que, em determinadas situações, pessoas se comportam como partículas em um sistema, com interações físicas e sociais que se assemelham a partículas carregadas ou a partículas em campos gravitacionais. Por conta disso, surgiram algumas propostas de modelos baseados em sistemas de partículas, onde cada agente na multidão é visto como uma partícula. Helbing e Molnár (1995) propõem um modelo onde os agentes se movem devido a

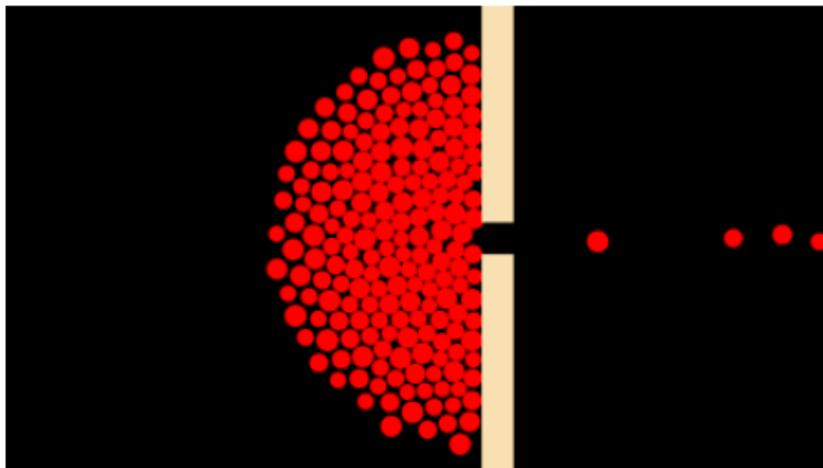
Figura 5 – Exemplo de modelo fazendo uso de regras. Buscar e fugir (esquerda). Perseguir e esquivar (direita).



Fonte: (REYNOLDS, 1999).

forças repulsivas exercidas por outros agentes, objetos, paredes e outros obstáculos dos quais os personagens devem manter distância, e forças atrativas causadas por amigos, objetivos e outros pontos de interesse dos quais o agente deve tentar se aproximar. Esse modelo consegue simular um efeito de auto-organização coletivo devido às forças estabelecidas, permitindo uma vasta customização dessas interações. Helbing *et al.* (2000) propõem um modelo baseado em partículas capaz de simular multidões em situações de pânico, combinando forças físicas e social-psicológicas. O algoritmo conseguia reproduzir muitos fenômenos observados, incluindo: efeitos de entupimento em gargalos e o correspondente aumento crescente da pressão, engarrafamentos em áreas abertas, efeito "rápido-é-devagar", uso ineficiente de saídas alternativas e início de pânico por contra-fluxo e impaciência.

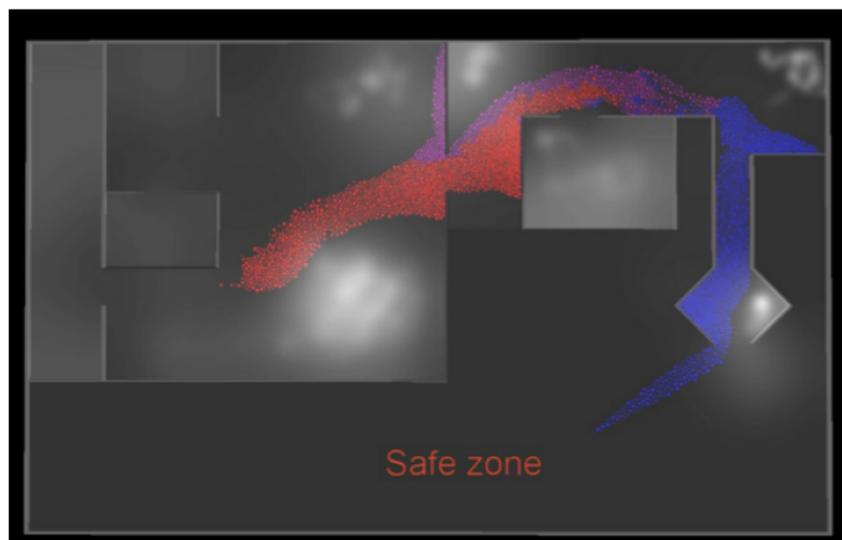
Figura 6 – Efeito de gargalo reproduzido por Helbing *et al.* (2000).



Fonte: Helbing *et al.* (2000).

Em (COURTY; MUSSE, 2005), uma força de repulsão foi adicionada ao modelo de Helbing *et al.* (2000) para fazer a multidão evitar lugares com fumaça enquanto tentam evacuar a construção (Figura 7). Em Braun *et al.* (2003), os autores estendem o modelo de Helbing para lidar com determinadas individualidades dos agentes e seus comportamentos a sós e em grupo. Modelos baseados em partículas também têm sido usados para estudar comportamentos em grupos e seus efeitos na dinâmica das multidões (MOUSSAÏD *et al.*, 2010; XU; DUH, 2010). Apesar de serem particularmente bons em simular situações de emergência, este tipo de modelo também pode ser adaptado para simular situações comuns, como as vistas em (PELECHANO *et al.*, 2007).

Figura 7 – Multidão sendo repelida por fumaça enquanto tentam evacuar a construção.



Fonte: Courty e Musse (2005).

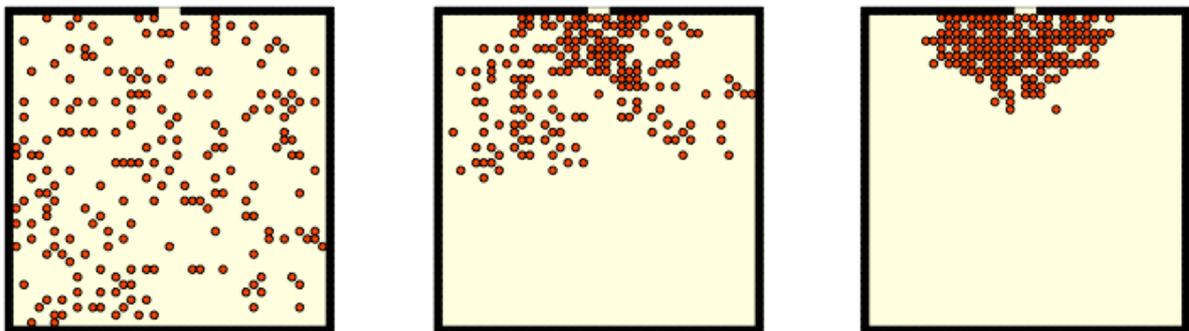
Uma abordagem diferente foi proposta por (HEIGEAS *et al.*, 2003). Neste caso, as interações dos agentes foram modeladas como um sistema massa-mola com amortecimento, onde o termo de elasticidade era modificado com respeito à distância relativa do agente. Modelos baseados em partículas são inerentemente reativos, fazendo com que os agentes não antecipem seus movimentos com relação a outros agentes em trajetória de colisão, resultando em artefatos visualmente desagradáveis em ambientes esparsos. Entretanto, é possível minimizar o efeito adicionando uma força evasiva para fazer com que o agente reaja a colisões futuras com antecipação, como proposto em (KARAMOUZAS *et al.*, 2009).

2.1.2.3 Modelos de autômato celular

Autômatos celulares são sistemas dinâmicos discretos em uma grade celular de duas dimensões. Uma célula pode estar ocupada, livre ou ter outro estado dependendo do comportamento a ser simulado. As regras são definidas de modo a controlar o comportamento dos agentes durante a simulação.

Os modelos de autômatos celulares (DIJKSTRA *et al.*, 2000; SCHADSCHNEIDER, 2001; TECCHIA *et al.*, 2001; BURSTEDDE *et al.*, 2001; KIRCHNER; SCHADSCHNEIDER, 2002; KIRCHNER *et al.*, 2003) são simples e eficientes, porém são incapazes de simular interação entre agentes, uma vez que cada célula da grade bidimensional só pode ser ocupada por um agente por vez, resultando em simulações não muito realistas em situações de alta densidade (Figura 8). Essas situações de alta densidade são importantes para garantir o realismo da simulação, essenciais em aplicações como filmes, jogos e realidade virtual. Pensando nisso, (LOSCOS *et al.*, 2003) adaptaram o modelo inserindo direções desvio de colisões em caso de multidões concentradas, acarretando na emergência de um fluxo entre os pedestres, permitindo a emergência de padrões locais e globais.

Figura 8 – Simulação de evacuação de uma sala usando modelo de autômato celular. É possível notar a organização dos agentes na grade regular.



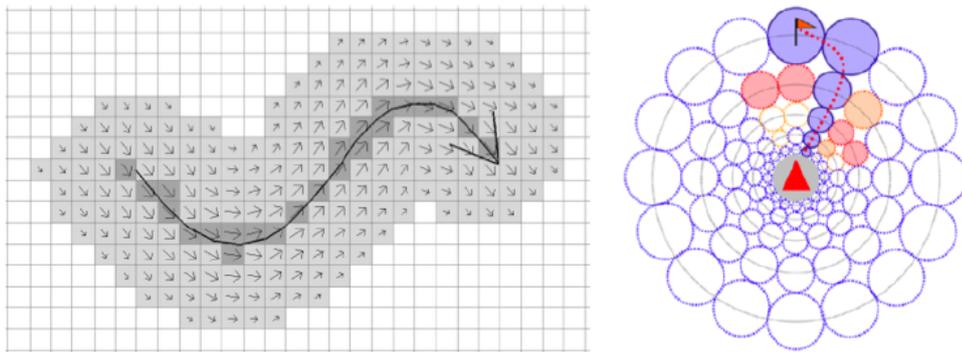
Fonte: Burstedde *et al.* (2001).

2.1.2.4 Modelos de campo de orientação

Modelos de campo de orientação podem ser usados tanto para planejamento local quanto global. O modelo consiste de regiões do cenário que possuem um campo que indicam a orientação e a velocidade que os agentes devem seguir. Em (CHENNEY, 2004), esses campos de orientação são definidos manualmente, criando ladrilhos de fluxo. Em (PATIL *et al.*, 2011),

o campo é gerado proceduralmente com algum nível de intervenção do usuário (Figura 9 (esquerda)). Seu trabalho também permite extração dos campos de fluxo a partir de vídeos de multidão. (KAPADIA *et al.*, 2009) usam campos de orientação locais para encontrar o melhor caminho possível para o agente de forma imediatista (Figura 9 (direita)). O problema dessa técnica é que isso levará o agente para um mínimo local.

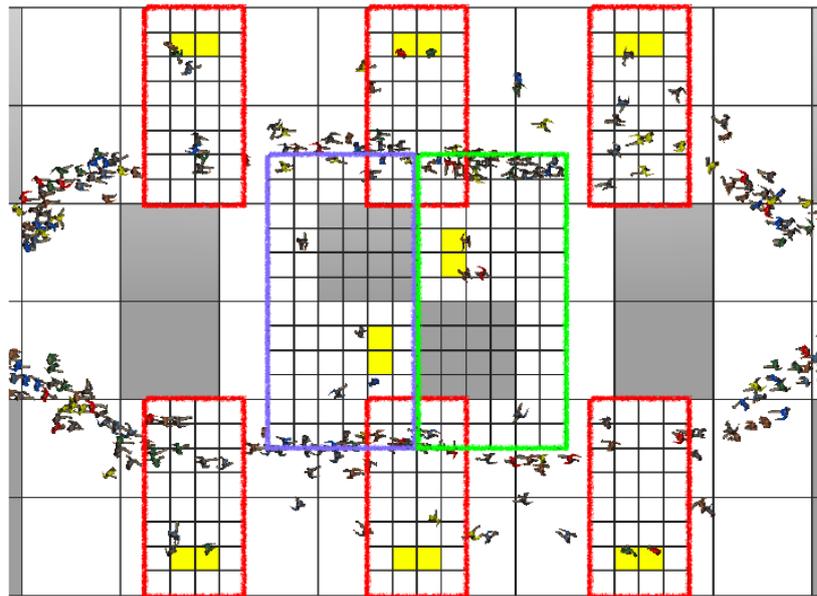
Figura 9 – Exemplos de campos de orientação. Campo atualizado por interação com usuário (esquerda) e campo egocêntrico (direita).



Fonte: Patil *et al.* (2011) (esquerda) e Kapadia *et al.* (2009) (direita).

Campos de orientação podem ser gerados a partir de campos potenciais (KHATIB, 1985; WARREN, 1989; WARREN, 1990; SHIMODA *et al.*, 2005). Em uma grade, as células contêm um valor potencial positivo que diminui de acordo com a proximidade do objetivo. Desse modo, células objetivo devem ter valor igual a zero, enquanto obstáculos devem ter valores elevados de modo que irão repelir os agentes. Um modelo onde equações diferenciais são usadas para definir a dinâmica da multidão foi proposto por (HUGHES, 2002; HUGHES, 2003). Neste modelo, os agentes são convertidos em um campo de densidade que são usados para gerar um campo potencial. Uma melhoria para este modelo foi proposta por (TREUILLE *et al.*, 2006), onde os agentes são separados em grupos com objetivos comuns e cada um desses grupos possui uma representação do ambiente em forma de uma grade de células, permitindo que, para cada grupo, seja calculado um campo potencial. (JIANG *et al.*, 2010) melhoraram o modelo de Treuille, permitindo ambientes mais complexos. (DUTRA *et al.*, 2013) propuseram um método de campo multipotencial, permitindo que múltiplos campos potenciais alterem momentaneamente o objetivo dos agentes (Figura 10).

Figura 10 – Exemplo de campo multipotencial. Oito campos potenciais são usados para mudar momentaneamente o objetivo do agente.



Fonte: Dutra *et al.* (2013).

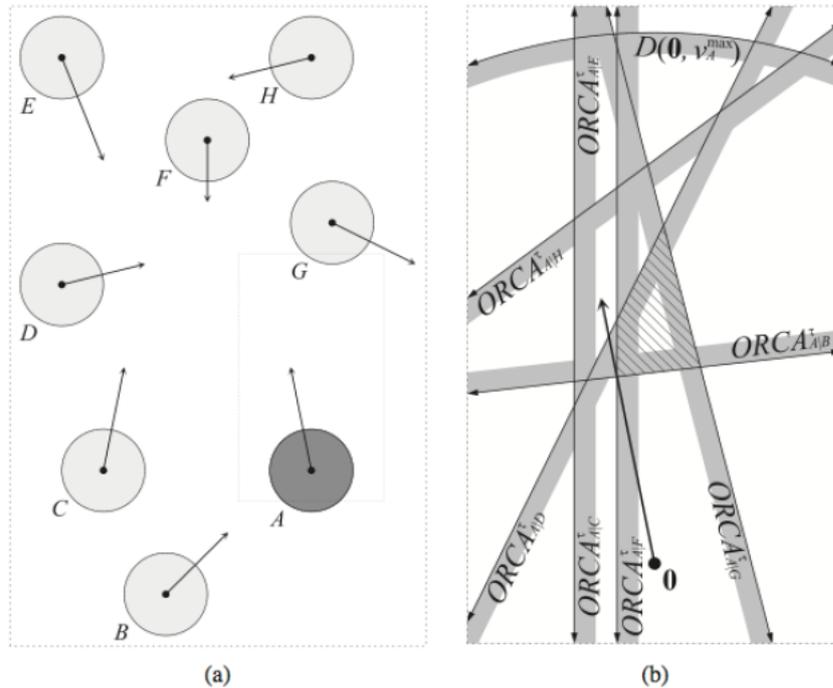
2.1.2.5 Modelos baseados em velocidade

Modelos baseados em velocidade são capazes de antecipar o perigo, extrapolando a trajetória dos agentes, baseado em sua velocidade atual, permitindo detectar possíveis colisões em um futuro próximo. (REYNOLDS, 1999) apresentou uma maneira de como agentes podem desviar de outros agentes que estejam em trajetórias arbitrárias, ou desalinhadas. A trajetória do agente é extrapolada, permitindo saber qual será a aproximação máxima entre dois personagens e se ela irá acontecer em um futuro próximo. (PARIS *et al.*, 2007) apresentaram um modelo que resolve determinados problemas do trabalho de Reynolds, como engarrafamentos e oscilações por conta da falta de antecipação. O modelo foi calibrado e validado com dados reais.

(BERG *et al.*, 2008) apresentaram o conceito de *Reciprocal Velocity Obstacle (RVO)*, ou Obstáculo de Velocidade Recíproca, que estende o conceito de Obstáculo de Velocidade (*VO*) usado no campo da robótica, apresentado por (FIORINI; SHILLER, 1998). Este modelo permite uma navegação segura e livre de oscilações entre os agentes considerando o comportamento reativo dos demais agentes, assumindo que agentes desviam uns dos outros da mesma maneira. (GUY *et al.*, 2009) apresentaram um algoritmo paralelo que usa um método de otimização discreto. Berg *et al.* (2011) propuseram uma técnica, denominada *Evasão de Colisão Ótima e Recíproca (Optimal Reciprocal Collision Avoidance ORCA)* que é capaz de computar, de modo eficiente, a solução ótima no espaço da velocidade, permitindo uma evasão recíproca

entre agentes, usando a maior velocidade próxima a velocidade de conforto que permite um movimento livre de colisões.

Figura 11 – (a) Uma configuração com oito agentes, onde suas velocidades são indicadas pelas setas. (b) Os semi-planos indicam as possíveis velocidades para o agente A, com respeito aos demais agentes. A seta indica a velocidade atual de A.



Fonte: Berg *et al.* (2011).

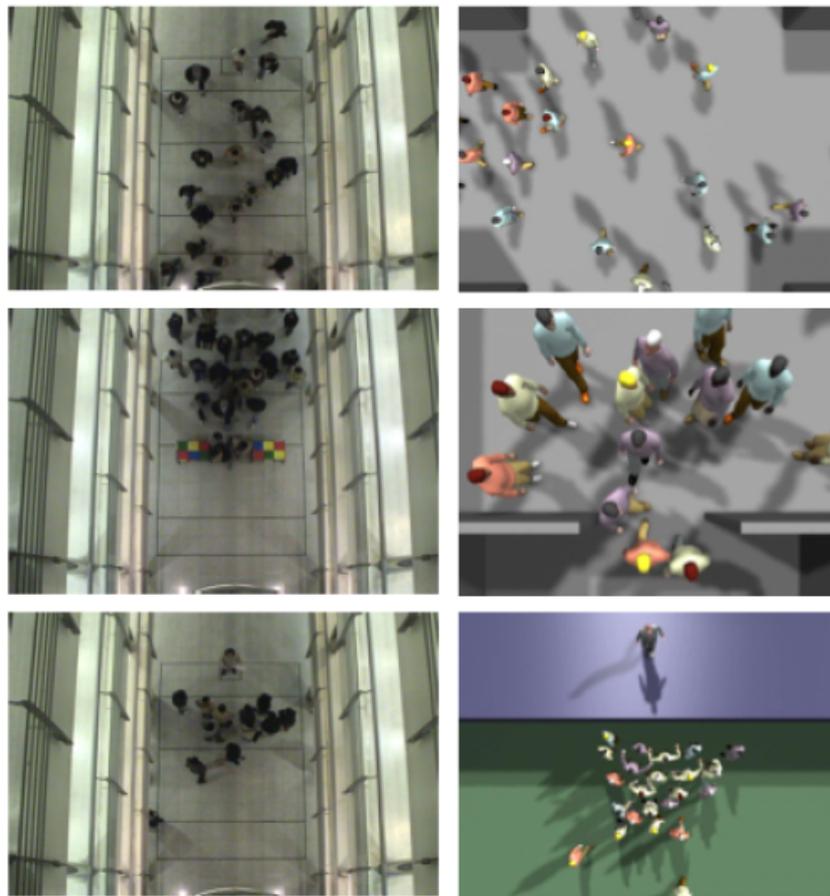
Esse modelo pode tratar casos de colisão entre vários agentes e obstáculos de modo eficiente, por ser um algoritmo inerentemente paralelo, permitindo simular multidões com trajetórias visualmente agradáveis em tempo real. Entretanto, essa qualidade decai à medida que a densidade da multidão cresce. Em situações simétricas, essa técnica pode apresentar artefatos.

2.1.2.6 Modelos orientados por exemplos

Modelos orientados por exemplos utilizam bases de dados para determinar o comportamento da multidão simulada como, por exemplo, dados de captura de movimentação de uma multidão real, trajetórias extraídas de experimentos, os resultados de outros modelos de simulação de multidão e até mesmo dados gerados pelo usuário. No modelo proposto por Metoyer e Hodgins (2004), o usuário pode ajustar a movimentação do agente em tempo real e, de acordo com essas interações, o simulador aprende como deve se ajustar a determinadas situações. Um modelo onde os padrões de comportamento são aprendidos a partir de filmagens

reais de multidões foi proposto por Lee *et al.* (2007). Lerner *et al.* (2007) propõem um modelo onde um conjunto de trajetórias extraídas de dados reais de multidões são usados para construir um banco de dados de exemplos de trajetórias, de modo a sintetizar incrementalmente essas trajetórias considerando os relacionamentos espaço-temporais de agentes e obstáculos próximos, e de acordo com cenários semelhantes encontrados nesse banco de dados.

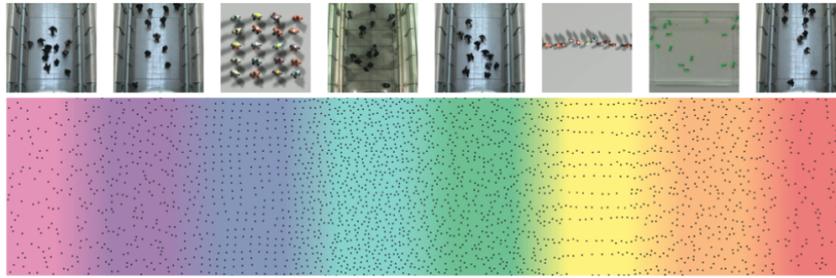
Figura 12 – Dados reais (esquerda) e comportamento reproduzido (direita).



Fonte: Lee *et al.* (2007).

Pettré *et al.* (2009) apresentaram um modelo que simula e reproduz trajetórias obtidas de diversas interações entre pessoas de uma multidão. Em Lerner *et al.* (2009), foi desenvolvido um método que é capaz de selecionar de um banco de dados o comportamento mais representativo da situação em que o agente se encontra atualmente. Paravisi *et al.* (2008) adaptaram o modelo de Treuille *et al.* (2006) para reproduzir comportamentos de grupos e de multidão extraídos de capturas de vídeo. Ju *et al.* (2010) apresentam um método que combina diferentes dados de várias multidões existentes para gerar uma nova multidão animada. Esses dados podem ser obtidos de dados reais ou de outras simulações de multidão (Figura 13).

Figura 13 – Dados reais (cima) e comportamento reproduzido (baixo).



Fonte: Ju *et al.* (2010).

Em Guy *et al.* (2011), os autores propõem uma técnica para gerar multidões heterogêneas a partir de estudos de percepção que afetam a personalidade dos agentes. Charalambous e Chrysanthou (2014) introduzem uma estrutura denominada *grafo de percepção-ação* (*perception-action graph* (PAG)), que recebe exemplos de multidão como um grafo, que é usado em tempo real para sintetizar comportamentos de multidões virtuais.

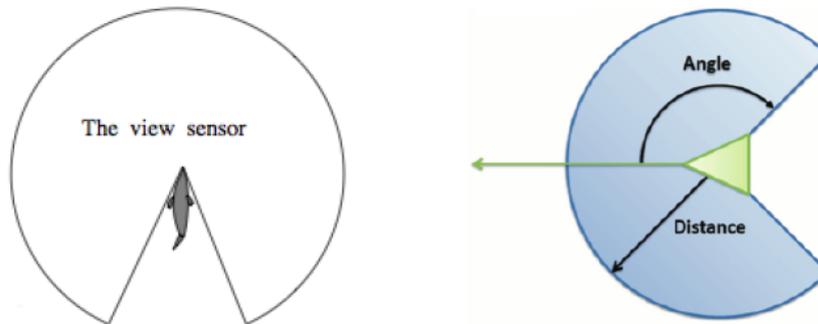
Esse tipo de modelo tem alguns problemas, como a necessidade de manter um banco de dados de comportamentos, que pode ocupar bastante espaço dependendo da quantidade de comportamentos desejados a se reproduzir. Isso pode também aumentar o tempo de busca por um comportamento específico. Além disso, a necessidade da entrada de dados estar de acordo com a situação a ser reproduzida, ou seja, o comportamento ter de ser gravado em uma situação específica que não pode ser usada em um contexto diferente pode dificultar o seu uso.

2.1.2.7 Modelos baseados em visão sintética

Algoritmos *baseados em Visão Sintética* (*Synthetic Vision* (SV)) equipam os agentes com uma câmera que é capaz de imitar a forma com que seres humanos percebem o mundo através do sentido da visão, e permitem que eles possam reagir ao risco de colisões futuras. Esses algoritmos são inspirados pelo reconhecimento da literatura do importante papel da visão na locomoção (CUTTING *et al.*, 1995; WARREN; FAJEN, 2004; RIO *et al.*, 2014). Tentativas iniciais de definir um campo visual usaram áreas geométricas (REYNOLDS, 1987; TU; TERZOPOULOS, 1994) ou representação volumétrica da cena (SILVA *et al.*, 2010) como representações do campo visual (Figura 14). Agentes devem reagir somente a objetos que estejam dentro de tais campos de visão.

A primeira simulação de locomoção dos agentes usando explicitamente a visão foi apresentada por Renault *et al.* (1990). Em Noser *et al.* (1995), os autores identificaram objetos

Figura 14 – Visão como uma área geométrica. Visão de um peixe (esquerda) e visão de um boid (direita).

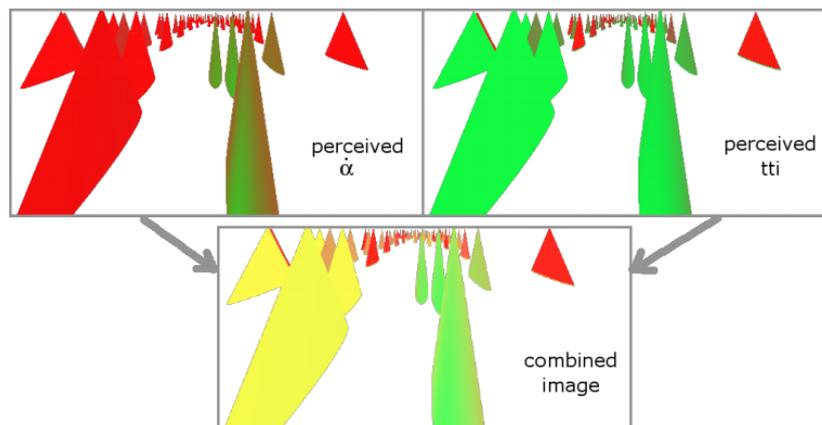


Fonte: Tu e Terzopoulos (1994) (esquerda) e Silva *et al.* (2010) (direita).

dentro de um ambiente usando a visão sintética. O trabalho de Peters e O'Sullivan (2003) usa a visão sintética para encontrar áreas interessantes no campo de visão do agente que possam chamar sua atenção, aumentando a sensação de presença dentro do ambiente virtual.

Ondřej *et al.* (2010) propuseram um novo modo de simular multidões com uso de visão sintética. Seu modelo consegue extrair informações que podem ajudar o agente a detectar riscos de colisão a partir de uma imagem capturada do ponto de vista de cada agente. Cada agente é capaz de tomar duas ações diferentes dependendo da situação em que se encontra. Ao detectar possíveis colisões futuras, o agente tentará girar em torno de seu eixo para mudar de trajetória. Caso uma colisão iminente esteja à sua frente, ele também é capaz de reduzir sua velocidade, evitando assim a colisão. Na Seção 2.2, mais detalhes sobre esse modelo serão apresentados.

Figura 15 – Visão de um agente no modelo proposto por Ondřej *et al.* (2010). As informações sobre a dinâmica dos obstáculos ($\dot{\alpha}$ e t_{ti}) são passadas para o agente através de seu sensor visual. A cor vermelha foi utilizada para representar valores pequenos.



Fonte: Ondřej *et al.* (2010).

Em Moussaïd *et al.* (2011), uma abordagem baseada em visão para simular comportamento de pedestres baseados em duas regras foi apresentado. Esse modelo é puramente reativo, do mesmo modo que o trabalho de Ondřej *et al.* (2010). A primeira regra é usada para adaptar a orientação do agente de modo a permitir um pequeno desvio do objetivo de acordo com a posição dos obstáculos, e a segunda regra adapta a velocidade de acordo com o tempo de reação do agente em relação ao primeiro obstáculo em sua trajetória. As colisões são evitadas usando um algoritmo baseado em sistema de partículas. Apesar de usar a visão do agente, o artigo não deixa claro se a visão é de fato uma imagem capturada ou uma representação geométrica do sentido.

Rio *et al.* (2014) investigam a informação visual usada para controlar a velocidade de pedestres que seguem outros pedestres. Uma lei, baseada na expansão ótica do seguidor em uma dimensão, mostra que o seguidor acelera caso o ângulo de visão do líder esteja diminuindo, e sua velocidade diminui caso o ângulo esteja aumentando e mantendo sua velocidade atual caso o ângulo seja mantido.

Esses modelos trazem um nível de realismo maior que simulações mais abstratas, pois aproximam a simulação das reais capacidades de multidões de extraírem informações de seu ambiente para locomoção. Esses resultados são compensados pelo alto custo computacional dessas técnicas. Espera-se que com a rápida evolução das placas de vídeo e técnicas de renderização em tempo real cada vez mais eficientes, essas técnicas possam se tornar populares das aplicações de tempo real, como jogos e realidade virtual.

2.1.2.8 Modelos híbridos

Cada modelo proposto possui suas respectivas vantagens e desvantagens. A ideia de combinar diferentes características dos modelos existentes deu origem aos modelos híbridos. Pelechano *et al.* (2007) capturam os melhores aspectos dos modelos baseados em regras e baseados em partículas, além de criarem um novo modelo que usa regras psicológicas, fisiológicas e geométricas, combinadas com forças físicas para reproduzir agentes heterogêneos e guiá-los pelo ambiente. Yersin *et al.* (2008) propõem uma arquitetura para simular multidões que divide o ambiente em regiões de interesse de acordo com a distância para a câmera, de modo que quanto mais próximo, mais importante. Em regiões que não possuem interesse, o planejamento é guiado por um grafo de navegação, e colisões não são evitadas. Em regiões com baixo interesse, o planejamento também é guiado por um grafo de navegação, e as colisões são evitadas baseando-

se nos conceitos definidos por Reynolds (1999). Regiões de alto interesse fazem planejamento de trajetória e evitam colisões usando campos potenciais, de modo semelhante a feita por Treuille *et al.* (2006).

Em Xiong *et al.* (2010), uma arquitetura que utiliza um modelo macroscópico e outro microscópico cooperam para conseguir os benefícios de cada um, quando necessário. O ambiente é particionado, e cada partição é regida por um modelo diferente dependendo da situação. Narain *et al.* (2009) apresentam um modelo baseado em campos potenciais, mas resolve o problema de colisões locais usando um modelo geométrico. Primeiro, o caminho do agente é planejado globalmente e sua velocidade de conforto é definida. Então, localmente, suas velocidades são adaptadas de acordo com a densidade da célula onde o agente se encontra. Depois de determinar a velocidade, uma distância mínima entre os agentes é garantida. Singh *et al.* (2011) descrevem um arcabouço que integra múltiplos modelos que são usados de acordo com a situação em que o agente se encontra. Golas *et al.* (2014) propõem uma abordagem que combina resultados de algoritmos discretos e contínuos, baseados em sua densidade local e na variação de velocidade.

Modelos híbridos conseguem lidar com uma quantidade mais variada de situações, dada sua flexibilidade de integrar diferentes soluções para diferentes problemas. Entretanto, quando se trabalha com diferentes modelos, é preciso levar em consideração a complexidade de integrar os modelos, para que eles possam trabalhar cooperativamente, sem interferirem no desempenho um do outro.

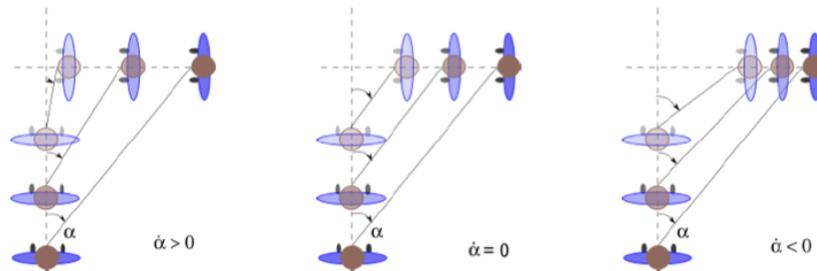
2.2 Modelo proposto por Ondřej

Nesta seção será detalhado o trabalho proposto por (ONDŘEJ *et al.*, 2010), pois foi um trabalho pioneiro dos modelos baseados em visão sintética. Nessa dissertação, melhorias sobre este modelo serão propostas, sendo então importante apresentar em detalhes a técnica proposta no trabalho. A ideia é compreender o seu funcionamento, seus pontos positivos e seus possíveis pontos fracos, para que então melhorias possam ser propostas.

O uso de visão sintética se propõe a imitar a forma com que o ser humano utiliza esse sentido durante sua locomoção e interação com o ambiente, permitindo que os personagens virtuais adaptem suas trajetórias para tornar sua movimentação mais crível. É conhecido pela literatura que a visão tem uma importância na locomoção (CUTTING *et al.*, 1995; WARREN; FAJEN, 2004; RIO *et al.*, 2014).

O trabalho de Cutting *et al.* (1995) afirma que seres humanos estão inconsciente-

Figura 16 – O ângulo de abertura α e sua derivada no tempo $\dot{\alpha}$ permitem a detecção de colisões futuras. Da perspectiva de um observador (personagem de baixo), uma colisão é prevista quando α não varia ao longo do tempo, ou seja, o valor de $\dot{\alpha}$ é 0 (centro). Caso o ângulo esteja variando ao longo do tempo, um dos dois agentes irá cruzar o ponto de colisão antes do outro, evitando o contato (esquerda, direita).



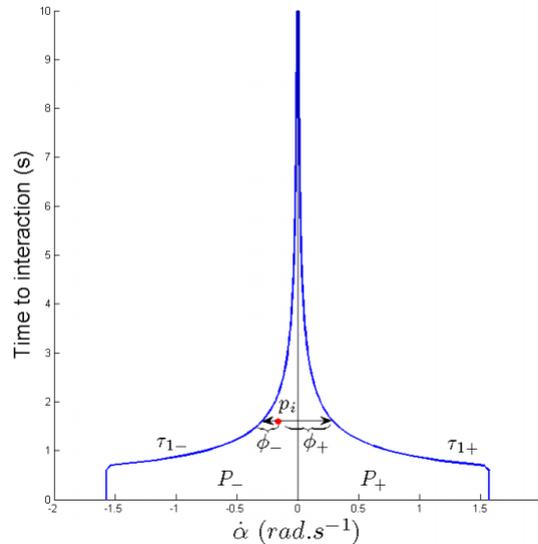
Fonte: Ondřej *et al.* (2010).

mente, mas constantemente, respondendo duas perguntas durante suas interações com o ambiente: *Uma colisão irá acontecer? Quando essa colisão irá acontecer?* O autor observou que essas perguntas poderiam ser respondidas usando a informação visual captada pelos agentes virtuais, através de indicadores. O primeiro indicador é o $\dot{\alpha}$, que representa a derivada no tempo do ângulo de abertura α (*bearing angle*) em que os obstáculos são percebidos (Figura 16). A segunda é um indicador de *tempo para colisão* (*time-to-collision* (*ttc*)), que é estimado a partir da taxa de crescimento dos obstáculos em sucessivas imagens percebidas pelo sistema visual. A informação relevante para se obter uma locomoção livre de colisões pode ser conseguida pela descrição do par $(\dot{\alpha}, ttc)$, sendo indicativo de colisão quando o valor de $\dot{\alpha}$ tende a zero e o valor de *ttc* baixo representando uma colisão iminente.

Baseando-se nesses conceitos propostos em Cutting *et al.* (1995), Ondřej *et al.* (2010) desenvolvem um modelo capaz de dirigir agentes virtuais de modo natural e desviá-los de colisões a partir do estímulo visual. Ondřej *et al.* (2010) adaptam o conceito de *ttc* para *tempo para interação* (*time to interaction* (*tii*)), onde o valor corresponde ao tempo exato para que aconteça alguma interação do observador com o obstáculo percebido.

Em seu modelo, uma câmera é posicionada sobre cada agente, renderizando uma imagem que corresponde à visão do avatar. Um par $(\dot{\alpha}, tii)$ é computado para cada pixel que compõe essa imagem, sendo que uma nova imagem é gerada a cada passo da simulação, com as posições dos agentes e dos obstáculos devidamente atualizadas. Cada pixel é avaliado pelo algoritmo como uma ameaça em potencial, sendo os valores do par analisados. Caso haja uma indicação de colisão futura ($\dot{\alpha}$ próximo de 0), o algoritmo define uma rotação para o agente, sendo essa o menor ângulo capaz de fazer com que ele desvie de todas as ameaças visíveis.

Figura 17 – A função de limiar definida por Ondřej *et al.* (2010). Há risco de colisão quando o par $(\dot{\alpha}, tti)$ está abaixo da linha da função, e com valores $tti > 0$. Também é possível ver que, quanto menor o valor de tti , maior será a reação do agente.



Fonte: Ondřej *et al.* (2010).

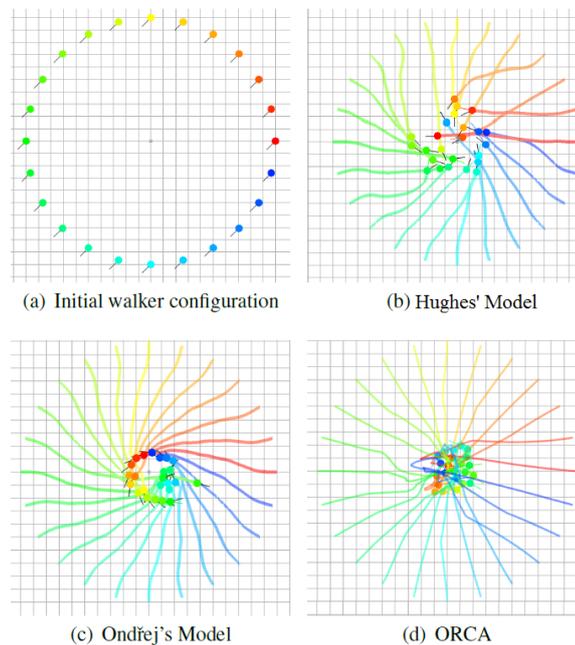
É imposto um limite pelo algoritmo, através de uma função de limiar (Figura 17), para que a movimentação se mantenha visualmente agradável. A segunda estratégia consiste em avaliar se há risco de colisão iminente (definido pelos autores como $tti < 3s$), a velocidade do agente é reduzida de acordo com o menor valor de tti encontrado dentre todos os pixels, para que o agente reduza sua velocidade somente o necessário para evitar uma colisão. Há também um limite de quanto o agente pode reduzir sua velocidade, de modo a evitar movimentos bruscos e irrealistas.

Apesar dos bons resultados, o algoritmo de Ondřej *et al.* (2010) possui alguns problemas. Grande parte é provavelmente consequência da maneira simples de responder aos estímulos visuais. Personagens deveriam, por exemplo, ser capazes de aumentar sua velocidade para escapar do obstáculo, e não apenas reduzir a velocidade. Como o modelo é reativo, colisões ainda são observadas com frequência. O agente se preocupa em desviar apenas do obstáculo mais próximo, descartando maior parte da informação captada pela visão. Isso faz com que ele negligencie algumas consequências de suas ações. As limitações das velocidades de rotação e redução de velocidade, apesar de tornarem a movimentação dos agentes virtuais mais verosímil, acabam acarretando colisões por ter sido detectada alguma ameaça iminente muito próxima ao agente ao acabar de ter entrado em seu campo visual, não dando tempo suficiente para o personagem reagir. Além disso, padrões de comportamento emergem em determinadas circunstâncias, o que é algo desejado em modelos de simulação de multidão, mas

sendo com intensidade muito elevada, tirando a naturalidade da movimentação dos personagens, possivelmente pelo fato de que todos os agentes se comportam de maneira muito semelhante.

2.3 Modelo adaptativo de densidade

Figura 18 – Comparação entre três modelos feitas em Hughes *et al.* (2015), em uma cena circular com 100 agentes. É possível notar que a aglomeração no centro da cena foi evitada graças à percepção de um aumento de densidade na multidão. Comparando com dados de captura, foi observado que pessoas tentam evitar situações de congestionamento.



Fonte: Hughes *et al.* (2015).

Em Hughes *et al.* (2015), é proposto um modelo para aprimorar os resultados encontrados por Ondřej *et al.* (2010). Nesse trabalho, é mostrada a importância da densidade em definir e adaptar a movimentação de pedestres em uma multidão. De forma semelhante, os valores de ttc_i e $\dot{\alpha}_i$ são calculados a partir do vetor de estado do agente para responder as duas perguntas relacionadas ao risco de colisão do agente. O controle do agente é feito sobre o par $(\dot{\theta}_i, v_i)$, que representam a velocidade angular e velocidade tangencial do agente, respectivamente. Baseando-se no trabalho proposto por Best *et al.* (2014), o valor da densidade do agente i , ρ_i também é calculado a cada passo da simulação. Esse valor é computado antes da imagem do agente ser gerada, sendo passado como uma constante para a GPU. Seguindo os mesmos passos do algoritmo de Ondřej *et al.* (2010), essa técnica identifica o pixel mais “perigoso” para definir

a ação a ser tomada.

Uma vez que o valor de ρ_i é computado, é calculado o *fator de densidade* (δ_i), simplesmente parametrizando esse valor com o uso de três constantes α_ρ , β_ρ e γ_ρ . O valor de δ_i é então usado nas fórmulas para determinar os novos valores de θ e v_i .

Esse modelo pode ser visto como um modelo híbrido, já que todo o cálculo da densidade é realizado antes do passo de visão sintética ter início. Pessoas só são capazes de perceber a presença de aglomerados dentro de seu campo de visão. Isso pode levar esse modelo a desconsiderar algumas limitações naturais da visão humana, criando situações visualmente desagradáveis.

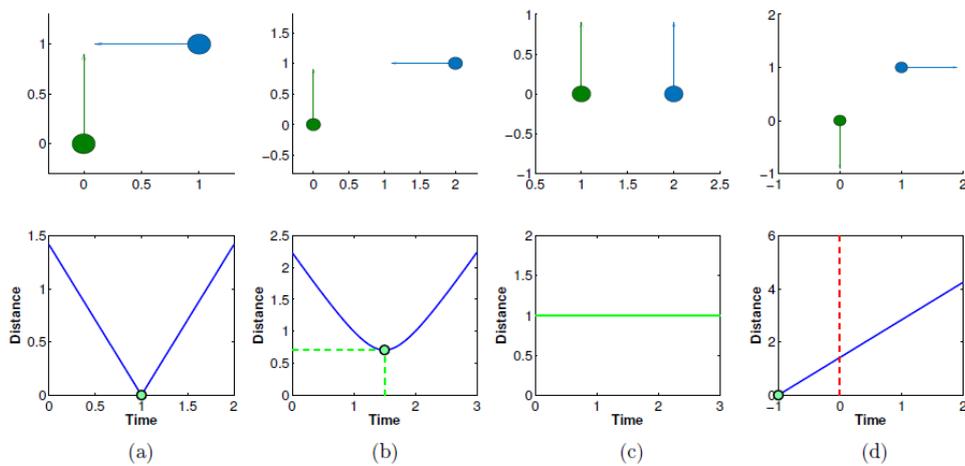
2.4 Modelo Baseado em Gradiente

Proposto por Dutra *et al.* (2015), esse modelo para controle de multidões virtuais se baseia na avaliação de uma função de custo em tempo real. Essa função de custo é responsável por, a partir da visão do agente, que também é uma imagem, calcular um valor, que corresponde ao risco de eventos indesejados acontecerem.

Em seu trabalho, primeiramente é mostrado que é possível definir uma função de custo para uma situação arbitrária. Baseando-se no trabalho de Cutting *et al.* (1995), a função de custo procura responder as mesmas perguntas. “*Uma colisão irá acontecer?*” e “*Quando uma colisão irá acontecer?*”. Para responder essas perguntas, é proposta a avaliação de índices análogos aos do trabalho de Ondřej *et al.* (2010), (α, ttc). Uma colisão em potencial é detectada a partir da *distância na aproximação máxima* (*distance at closest approach* (dca)). Uma possibilidade de colisão é detectada caso o valor de dca seja maior que zero e menor que a distância máxima permitida entre dois agentes, possivelmente a soma de seus respectivos raios. Se suas trajetórias forem mantidas, é só questão de tempo até que a colisão ocorra. A resposta para a segunda pergunta é obtida calculando o tempo para que a distância em dca seja atingida, ou seja, o *tempo para a aproximação máxima* (*time to closest approach* ($ttca$)). A Figura 19 ilustra diferentes situações e seus respectivos valores de dca e $ttca$.

Dutra *et al.* (2015) definem o estado atual do agente a pela sua posição p_a , orientação θ_a e velocidade escalar s_a , que são usadas para calcular sua velocidade v_a . A partir desses estados, é possível calcular os valores de $dca_{o_i|a}$ e $ttca_{o_i|a}$, que são os indicadores relativos ao obstáculo o_i tendo como observador o agente a . Com esses valores encontrados, é possível encontrar o custo que aquele obstáculo representa para o agente a .

Figura 19 – As figuras ilustram quatro situações diferentes com dois agentes. A linha superior mostra as posições dos agentes e suas velocidades, e a linha inferior, a distância entre os agentes ao longo do tempo. Na linha inferior, o ponto verde no gráfico indica o par $(dca, ttca)$ para cada uma das situações. (a) e (b) representam situações onde os agentes estão convergindo, em (c) os agentes se movem paralelamente e em (d), as trajetórias dos agentes divergem. Em (a), o valor de dca é 0 com $ttca$ em 1, o que indica que em 1 segundo acontecerá uma colisão entre os dois agentes. Em (b), o valor de dca indica que não há um risco de colisão, pois seu valor é maior que a soma dos raios dos dois agentes. Em (c) o valor de dca é constante e o valor de $ttca$ é indeterminado, pois os agentes não estão nem se aproximando e nem se afastando, mantendo uma distância constante. Em (d), o valor de $ttca$ é negativo, indicando que os pontos em que as trajetórias dos agentes se encontram estão na direção oposta da qual pelo menos um deles está indo.

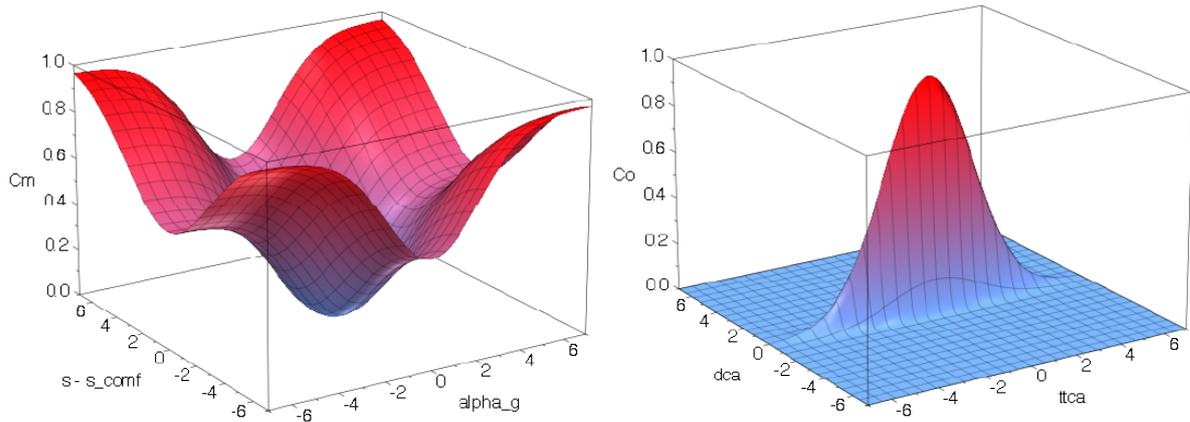


Fonte: Dutra *et al.* (2015).

Dutra *et al.* (2015) definem uma função de custo C_t como sendo a soma de duas outras funções mais especializadas, C_m e C_o . C_m , é uma função que calcula o custo relacionado ao objetivo do agente e sua velocidade. Essa função (Figura 20, esquerda) será mínima quando o agente estiver indo em direção ao seu objetivo (Figura 20, esquerda, $\alpha_g = 0$) em sua velocidade de conforto (Figura 20, esquerda, $s - s_{comf} = 0$), e aumenta a medida que se afasta. A outra função, C_o , trata do custo relacionado aos obstáculos. O valor dessa função será mínimo quando não houver risco de colisão, e à medida que esse risco aumenta essa função computa valores mais elevados (Figura 20, direita). O custo $C_{o_i|a}$ é definido como uma Gaussiana de duas dimensões, levando em conta os valores de dca e $ttca$. O valor do custo de obstáculos total, C_o , é uma média aritmética dos custos $C_{o_i|a}$ de todos os obstáculos i .

Cada agente se move em direção ao seu objetivo evitando riscos de colisão com obstáculos. Como a função C_t modela esses dois critérios, os agentes continuamente adaptam suas velocidades e orientações para otimizar o custo de maneira local. Em Dutra *et al.* (2015), isso é feito computando o vetor gradiente da função C_t , que aponta no sentido em que a função

Figura 20 – (esquerda) Função de custo de movimento C_m , definida como a soma de duas funções Gaussianas relativas ao ângulo do agente em relação ao seu objetivo e sua velocidade atual em relação a sua velocidade de conforto. (direita) Plot da função de custo C_o em relação aos valores de dca e $ttca$.



Fonte: Dutra *et al.* (2015).

diminui com maior intensidade, e atualizando a velocidade e a orientação do agente para seguir nessa direção.

Todas essas etapas são computadas para cada agente, em cada passo da simulação, de forma semelhante como é feita por Ondřej *et al.* (2010). Cada obstáculo i é representado por um pixel da imagem capturada pelo sensor visual do agente a . Porém, diferente do outro modelo, Dutra *et al.* (2015) utilizam toda a informação contida no campo visual, o que permite uma maior antecipação de riscos após sua movimentação, pois leva em consideração tudo que o agente é capaz de enxergar em seu campo de visão, no lugar de considerar apenas o obstáculo que representa maior risco.

2.5 Considerações finais

Esse capítulo começou com uma visão geral do que é planejamento global e planejamento local. Foi feita então uma pequena introdução a algumas técnicas de planejamento local, explicando brevemente suas características e diferenças.

Três técnicas foram apresentadas com maiores detalhes. A seção 2.2 apresentam um modelo de simulação de multidões que inova por usar informação capturada de uma renderização representando a visão do personagem. A 2.3 apresentou uma técnica que expande o primeiro modelo adicionando a noção de percepção de densidade da multidão, com o intuito de evitar aglomerações de agentes que possam levar a colisões ou travamentos. A seção 2.4 expandiu o conceito do primeiro trabalho mudando a forma com que os dados capturados são utilizados,

transformando um algoritmo que considerava apenas os casos mais arriscados por uma técnica que calcula um valor de custo baseado em toda a imagem.

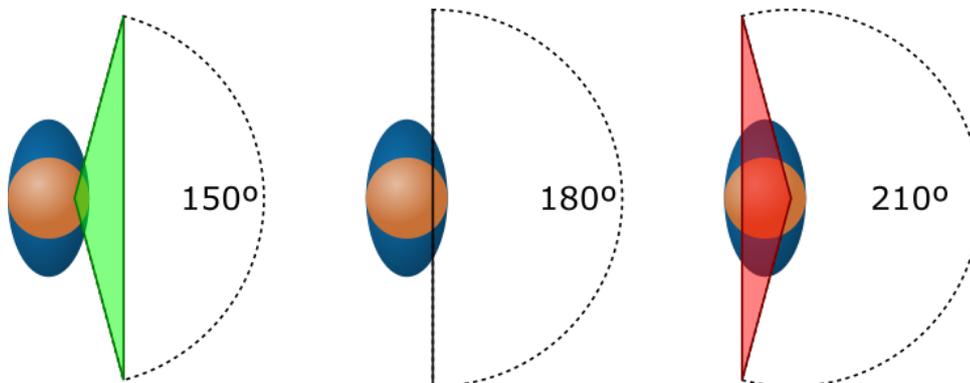
No próximo capítulo, uma nova técnica para gerar a renderização da visão dos agentes é apresentada. Nessa técnica, o agente ganha a capacidade de ver a cena de forma panorâmica, permitindo ângulos de visão maiores que 180° .

3 VISÃO PANORÂMICA

3.1 Campo de visão em modelos baseados em visão sintética

O algoritmo proposto por Ondřej *et al.* (2010), que daqui em diante será referido como OSV (Ondrej Synthetic Vision), e conseqüentemente os modelos propostos a partir de sua técnica, supõe que a visão dos agentes funciona como uma câmera *pinhole*. O campo de visão é definido por um *frustum*, ao qual se associam projeções planares, ou seja, a imagem será projetada sobre um plano bidimensional. Como é possível perceber pela Figura 21, o campo definido desta forma fica limitado geometricamente a ângulos menores que 180° . Além disso, distorções começam a aparecer conforme o ângulo aumenta. Essas distorções acontecem à medida que objetos se aproximam das bordas do *frustum*, que ao serem projetados em um plano, são esticados. E tornam-se mais visíveis a medida que o ângulo se aproxima de 180° .

Figura 21 – Exemplos de campos de visão com *frustum* de projeção sendo observados por cima da cabeça dos avatares. A imagem da esquerda mostra o exemplo com um ângulo menor que 180° . Na imagem do meio, um exemplo com ângulo de 180° . Nesse caso os planos *near* e *far* coincidem, ou seja, o frustum não possui volume o que faz com que nada seja projetado no plano e conseqüentemente a cena não seja renderizada. A imagem da direita mostra um exemplo com um ângulo maior que 180° . Nesse caso, o problema é que o plano *far* fica atrás do plano *near*.



Fonte: Autor

O modelo OSV utiliza uma projeção planar com campo de visão menor que 180° de abertura horizontal (Figura 21 (esquerda)), mais precisamente com o valor de 150° , e com ângulo vertical de abertura de 80° , o que limita o alcance do modelo não só para a simulação de humanos como para a simulação de outros animais (várias espécies de pássaros, por exemplo, possuem campo de visão com abertura horizontal maior que 180° (MARTIN, 1994)).

Também foi observado que em alguns experimentos, agentes colidem com obstáculos

próximos logo após uma mudança de trajetória brusca. A suspeita principal era a falta de predição das consequências de sua movimentação. O modelo de Dutra *et al.* (2015) aprimorou este modelo adicionando uma função de custo que dá ao agente um maior poder de antecipação de colisões graças à continuidade de sua função de custo. Mesmo neste modelo, colisões continuam acontecendo. Para averiguar se as causas de tais colisões eram de fato culpa da limitação de seu campo de visão, a técnica de visão panorâmica foi proposta.

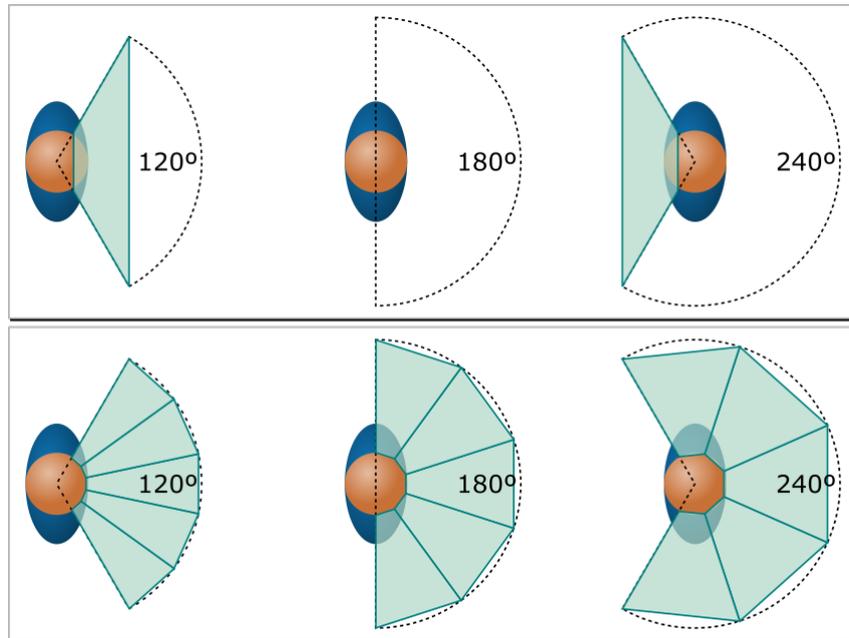
3.2 Técnica Proposta

Essa limitação do campo de visão dos agentes motivou a adição de uma visão panorâmica ao modelo. Panoramas se tornaram populares no século 19 como forma de representar paisagens inteiras a partir de um ponto de vista. Com o avanço da tecnologia, câmeras fotográficas e até mesmo celulares passaram a adotar softwares para geração de imagens panorâmicas. Apesar de existirem várias técnicas patenteadas para captura de fotografias panorâmicas, todos os softwares funcionam de um modo semelhante, compondo a imagem panorâmica a partir de múltiplas fatias de fotografias comuns, onde cada uma é retirada a partir de um ângulo diferente. A ideia para a implementação baseou-se em uma modificação chamada *PanQuake* (OORTMERSSSEN, 2015), que muda a câmera do jogo de tiro em primeira pessoa *Quake* (*id Software*) para uma câmera panorâmica cilíndrica. A abordagem utilizada consiste em renderizar finas imagens do ambiente a partir de diferentes pontos de vista, rotacionando-se a câmera em torno do eixo y, para no final ter uma renderização completa de um panorama horizontal, semelhante ao que é feito com câmeras panorâmicas reais. O resultado deste processo são os diferentes *frustums* de projeção ilustrados na Figura 22.

3.3 Implementação

Para descrever o campo de visão panorâmico segundo esta técnica, são necessários dois parâmetros adicionais. O primeiro se refere ao ângulo do campo de visão horizontal desejado para a projeção, e o segundo se refere à quantidade de fatias (renderizações individuais) utilizadas para definir o panorama. Uma das etapas da execução de um algoritmo baseado em visão sintética é o posicionamento da câmera “na cabeça do agente”. Agora, como pode ser visto no Algoritmo 1, após o posicionamento é necessário realizar s renderizações (uma para cada fatia) (da linha 2 até a linha 11) ajustando o *viewport* (linha 4) e o *frustum* de projeção (linha 6

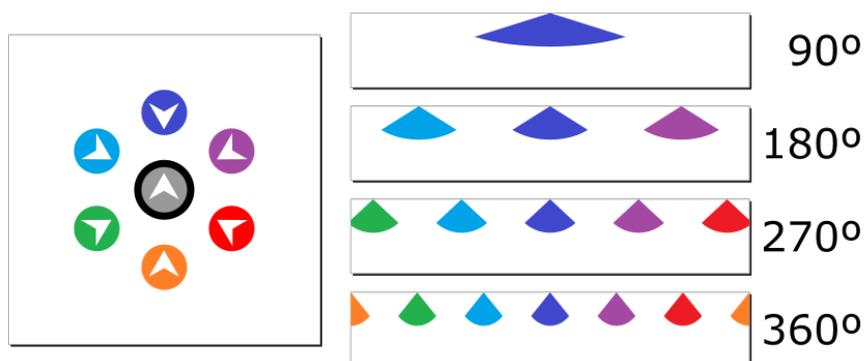
Figura 22 – Visão de cima, comparando diferentes ângulos de visão horizontal (120° , 180° , 240°), com 1 e 5 fatias, respectivamente.



Fonte: Autor.

até a linha 9) para cada nova renderização, de forma a encaixar todas elas em uma única imagem (linha 10). A função utilizada na linha 10 funciona da mesma forma que nos dois modelos citados anteriormente, sendo aqui chamada para cada fatia. Por fim, as funções tanto da linha 12 como da linha 14 são executadas de forma idêntica ao modelo original. Na Figura 23 pode-se ver o resultado da renderização fazendo uso deste algoritmo. No Apêndice A, é detalhado o cálculo realizado pelo algoritmo e como ficam as diferentes fatias de renderização após serem devidamente dimensionadas e rotacionadas.

Figura 23 – Na imagem, a visão do agente em destaque ao centro é reproduzida para diferentes campos de visão no cenário circular exibido à esquerda. As imagens da direita representam a visão do agente para um ângulo horizontal variando de 90° a 360° , com um ângulo vertical fixo de 80° .



Fonte: Autor.

Algoritmo 1: Implementação da visão panorâmica.

```

1: for all agent  $a$  do
2:     for all slice  $s$  do
3:          $sliceWidth \leftarrow textureWidth \div numberOfSlices$ 
4:          $setViewport(s \times$ 
5:              $sliceWidth, 0, sliceWidth, textureHeight)$ 
6:          $sliceFoV \leftarrow fovX \div numberOfSlices$ 
7:          $projection \leftarrow$ 
8:              $createFrustum(nearValue, farValue, sliceFoV, fovY)$ 
9:
10:         $rotY \leftarrow 0.5 \times (-fovX + (2 \times s + 1) \times sliceFoV)$ 
11:         $projection \leftarrow rotateY(projection, rotY)$ 
12:         $setProjectionMatrix(projection)$ 
13:         $texture(s) \leftarrow renderVisionToTexture()$ 
14:    end for
15:     $processVision(texture)$ 
16: end for
17:  $doModelSecondLoop()$ 

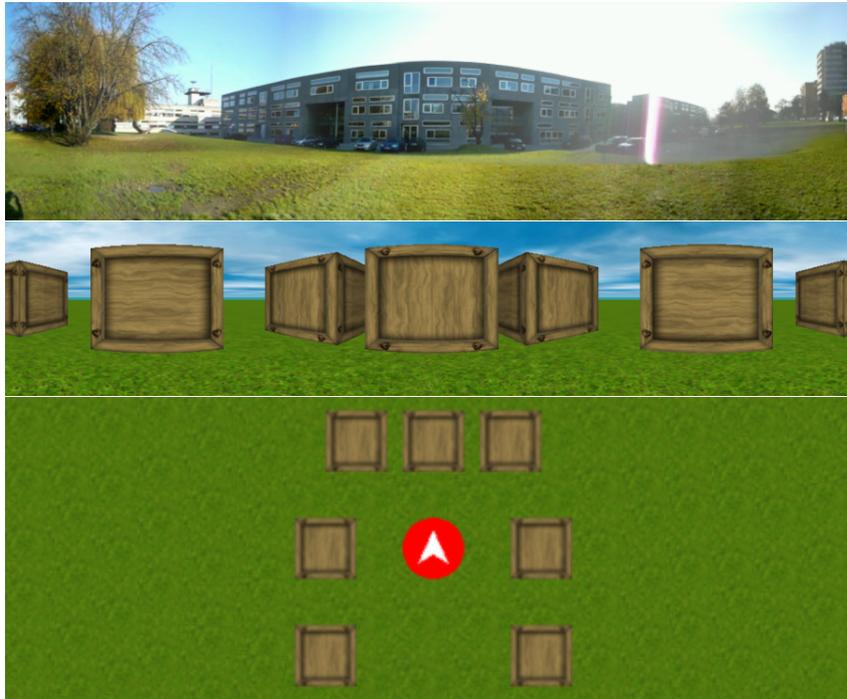
```

A modificação necessária é feita no momento da renderização do ambiente do ponto de vista do agente. Portanto, todos os modelos baseados em visão sintética como o OSV e também o modelo proposto por Dutra *et al.* (2015) podem usufruir desta modificação. Na Figura 23, são exibidos alguns resultados da modificação, onde a visão de um agente é ilustrada para diferentes campos de visão, no passo inicial de um cenário circular (Figura 23 (esquerda)).

3.4 Resultados

Após a realização de diversos experimentos com a inclusão da visão panorâmica nos modelos, foram observadas as mudanças causadas no comportamento dos agentes, o modo como a técnica afetou o desempenho da aplicação, e como diferentes valores dos parâmetros afetam cada uma dessas características.

Figura 24 – Na primeira imagem vê-se um panorama real. Na segunda imagem, um panorama renderizado utilizando a técnica, com ângulo de 270° . Na terceira imagem, a cena renderizada na segunda imagem vista por cima do agente.



Fonte: Google Images (topo) Autor (meio, baixo).

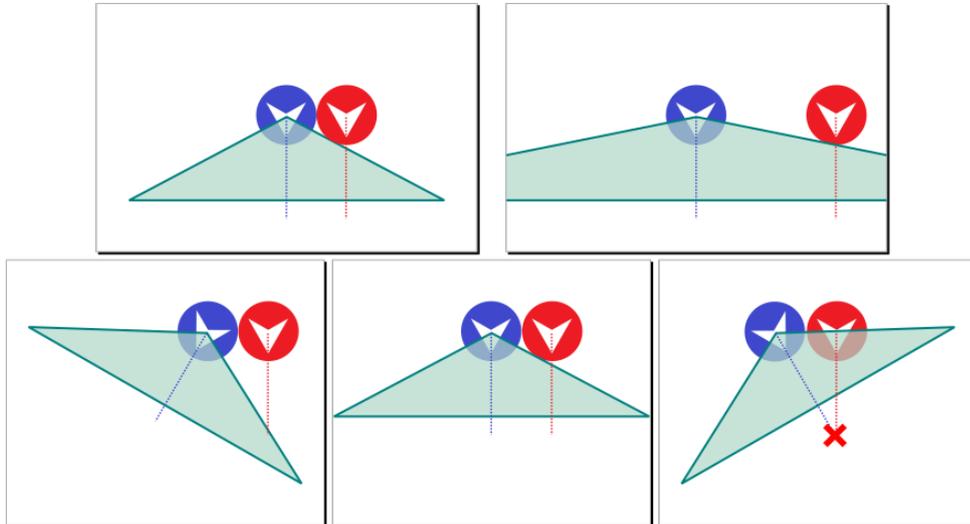
3.4.1 Visão panorâmica

Ao inserir a visão panorâmica no OSV, foi feito um estudo para verificar a influência que os diferentes parâmetros que modificam o campo de visão do agente afetam o resultado. Os parâmetros estudados foram: *ângulo horizontal*, *ângulo vertical*, *orientação vertical* e o novo parâmetro inserido pela técnica, o *número de fatias de visão*.

Um dos principais objetivos de aumentar a capacidade visual dos agentes foi para evitar que colisões indesejadas acontecessem por questões de falta de visibilidade. Foi observado que o menor ângulo de visão necessário para que dois agentes se vejam mutuamente é calculado a partir da configuração ilustrada na Figura 25.

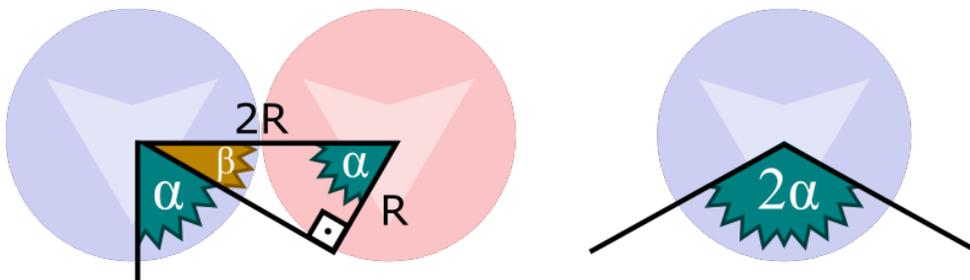
Em seu trabalho, Ondřej *et al.* (2010) usou um ângulo de 150° para abertura horizontal, o que já garante aos agentes uma visibilidade suficiente de seu ambiente para que possam desviar dos demais agentes, por ser maior que o ângulo mínimo necessário para garantir visibilidade (Figura 26). Os valores do ângulo vertical e da orientação vertical são de 80° e -40° , respectivamente. Os valores fazem com que o plano superior do *frustum* fique horizontal, uma vez que não há em seus cenários obstáculos acima do nível da cabeça dos agentes. O plano inferior fica quase tangente à inclinação do agente, permitindo que o mesmo tenha uma visão de

Figura 25 – Menor ângulo necessário para que dois agentes possam se ver. A situação ilustrada é o caso extremo em que dois agentes estão colados lado a lado e qualquer alteração de rotação que possa causar uma colisão fará com que os agentes se vejam antes. 2α representa o valor total de abertura do campo de visão.



Fonte: Autor.

Figura 26 – Ilustração mostrando como é calculado o menor ângulo com as considerações da Figura 25. Sabendo que o agente possui raio R , e ao estar lado a lado com outro agente a distância entre seus centros é de $2R$, podemos calcular o ângulo mínimo de visão 2α . O valor de α é $\arccos(\frac{R}{2R}) = 60^\circ$. Portanto o ângulo de visão deve ser no mínimo 120° .



Fonte: Autor.

objetos que estejam bem próximos.

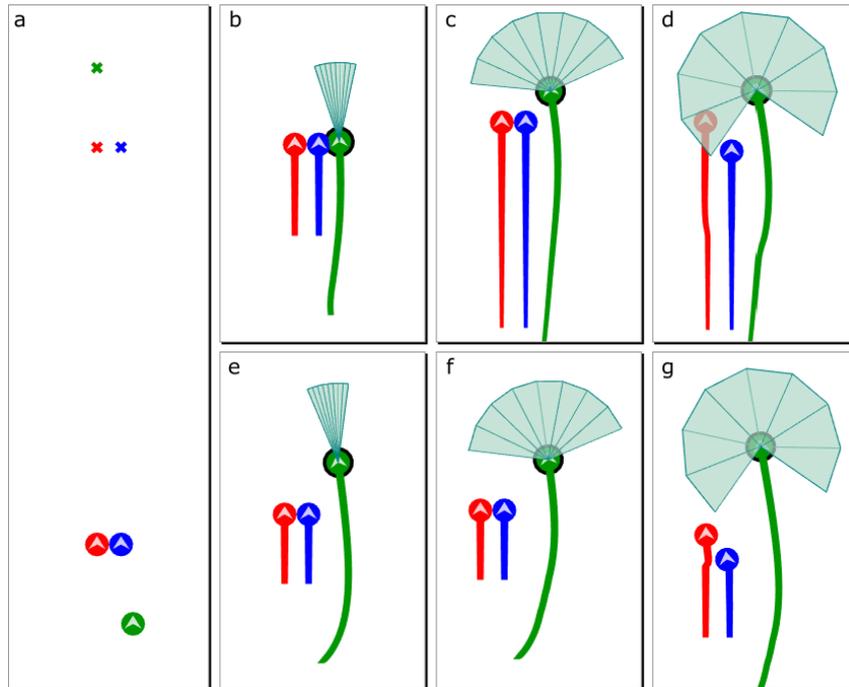
Aumentar o campo de visão dos agentes, entretanto, causou mudanças de comportamento. Os algoritmos passaram a receber um maior volume de visualização para sua entrada, permitindo reagir a eventos que antes não eram detectados.

3.4.2 Comportamento dos Agentes

Foi estudado o impacto da variação dos parâmetros que definem o campo de visão sobre os modelos de Ondřej *et al.* (2010) (OSV) e Dutra *et al.* (2015), aqui em diante referido como GBM (Gradient-based Model). Na Figura 27 observa-se a variação de comportamento do

agente para diferentes valores de ângulo de visão horizontal.

Figura 27 – Alguns testes realizados com o campo de visão variando o ângulo horizontal. Em (a), a configuração do cenário. O agente vermelho e o agente azul caminham lado a lado, paralelamente, enquanto o agente verde, mais rápido, tenta uma ultrapassagem. Posição inicial dos agentes e seus objetivos, marcados por quadrados de suas respectivas cores. Em (b,c,d) é utilizado o modelo OSV, e em (e,f,g) é utilizado o modelo GBM. Em (b,e) o ângulo de 30° é utilizado. Em (c,f) o valor é de 150° . Para (d,g) o valor é de 270° .



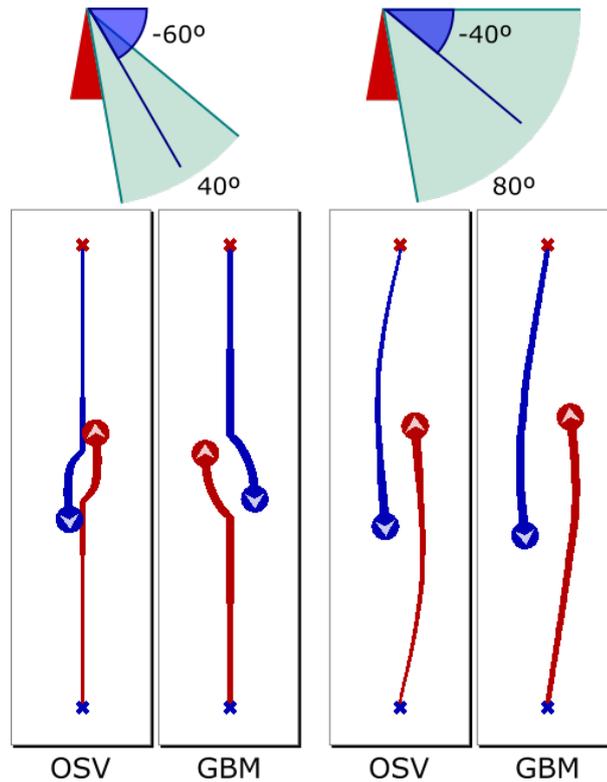
Fonte: Autor.

É possível ver que para ângulos menores que 180 graus (Figuras 27 (b), (c), (e) e (f)), os agentes vermelho e azul não reagiram à chegada do agente verde, enquanto nos demais casos, ao perceber a chegada de uma ameaça muito rápida vindo por trás, esses agentes tomaram medidas evasivas, diminuindo suas próprias velocidades e fazendo leves desvios para o lado. Com ângulos menores que o ângulo mínimo de 120° , o agente verde deu início à sua manobra de ultrapassagem muito cedo, o que causou uma colisão com os outros dois agentes. Isso aconteceu por não ser capaz de vê-los ao saírem de seu campo de visão, o que indicou para seu cérebro que o caminho estava livre.

Essa simulação nos mostra que é possível imitar o comportamento de diversos tipos de agentes diferentes, com os mais variados campos visuais. Também mostrou que o ângulo do campo de visão é um fator importante para evitar colisões. Mas valores muito altos para o campo de visão, até mesmo próximos de 360 graus podem afetar a naturalidade da simulação.

Seres humanos não são capazes de enxergar completamente seus arredores, o que pode causar estranheza da simulação para observadores humanos.

Figura 28 – Estudo do ângulo vertical e da orientação da visão do agente. Na esquerda, são testados nos dois modelos um ângulo de visão reduzido, de 40° de abertura e -60° de orientação. Na direita, são testados os valores padrão dos modelos, com abertura de 80° e orientação de -40° .

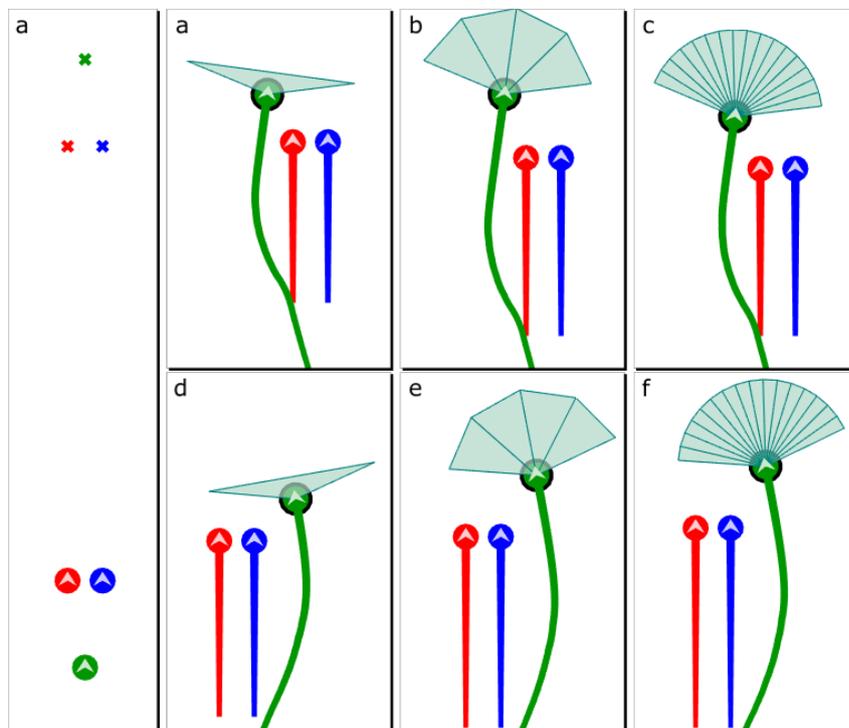


Fonte: Autor.

Também foi estudado o impacto da variação no ângulo vertical (Figura 28). Foi observado que, por causa do fato dos modelos atuais não utilizarem a informação no sentido vertical, uma vez que toda a movimentação dessas simulações acontece em duas dimensões, alterar o campo no sentido da vertical tem apenas o poder de influenciar na antecipação do agente. Com as configurações de ângulo definidas por Ondřej *et al.* (2010), o plano inferior do volume de visualização é tangente à superfície do personagem observador, e o plano superior é horizontal, se situando acima do nível da cabeça do agente. Isso quer dizer que tudo que se encontra, no plano bidimensional, contido no ângulo de visualização horizontal será enxergado pelo personagem. Modificando esses ângulos de forma a simular que o agente esteja olhando para baixo, foram definidos os outros valores de ângulo apresentados na Figura 28. O que foi observado no comportamento do agente foi uma redução na sua capacidade de antecipação. Por demorar mais a enxergar os obstáculos, o agente demora mais para realizar manobras evasivas.

Por fim, foi estudado o impacto que o número de fatias causa no agente. Como visto anteriormente na descrição da técnica, ficou claro que a renderização panorâmica usando fatias não gera uma imagem igual àquela que seria obtida caso fosse utilizada uma única câmera *pinhole* com projeções planares. Então é esperado que hajam variações de comportamento nos agentes. Na Figura 29 é ilustrado um exemplo de experimento realizado com diferentes números de fatias.

Figura 29 – Estudo do impacto do número de fatias no comportamento do agente. Mesmo a imagem resultante da renderização sendo diferente, pouca variação foi observada na maioria dos testes. Em (a), a configuração inicial, semelhante à Figura 27. Em (b,c,d) é testado o modelo OSV, enquanto em (e,f,g) é testado o modelo GBM. Em (b,e) os modelos são testados com sua visão padrão, com apenas uma fatia. Em (b,e) é testado com 4 fatias, e em (c,f) com 16 fatias.



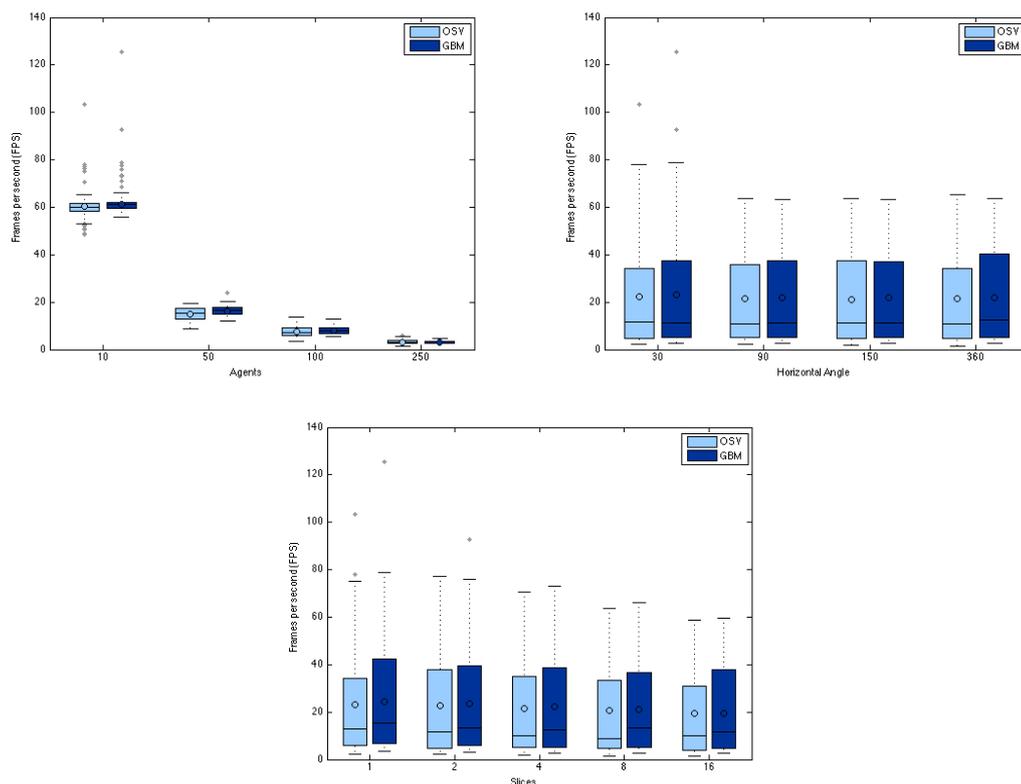
Fonte: Autor.

Mesmo com a diferença na renderização final, na maioria dos casos testados houve pouca variação da trajetória dos personagens. Como a variação acontece em todos os passos da simulação, é de se esperar que pequenas variações possam causar mudanças maiores na trajetória caso a simulação seja realizada em um caminho maior. Mesmo com essas variações de comportamento, semelhante a obedecendo às regras impostas pelo modelo, que evitam sua colisão com obstáculos e o levam a atingir seu objetivo.

3.4.3 Desempenho

Além de estudar o impacto no comportamento dos agentes, foi feito um estudo sobre o impacto no desempenho, uma vez que há um custo adicional para computar as novas projeções, e o *overhead* de realizar várias renderizações em várias fatias. Foi considerado o valor de *FPS* (quadros por segundo) em diferentes situações. Foram variados o número de agentes, número de fatias de renderização, ângulo de visualização horizontal e vertical e orientação vertical. Foram testados em cenários diferentes também, como no cenário circular e cruzamento. Na Figura 28 são ilustrados os gráficos dos três parâmetros que causaram maior variação no valor da *FPS*

Figura 30 – O resultado do teste de desempenho. O primeiro gráfico representa a *FPS* para o número de fatias. O segundo gráfico a influência do ângulo horizontal na *FPS*. E na terceira imagem, é analisado o número de fatias na renderização. Em azul claro encontram-se os resultados obtidos com o modelo OSV, e em azul escuro, os resultados obtidos com o modelo GBM.



Fonte: Autor.

Na comparação de ângulo horizontal, nenhuma correlação com a *FPS* foi encontrada. Dada a forma que o OpenGL renderiza a cena, era um resultado já esperado. O número de *texels* do *buffer* se manteve igual ao proposto por Ondřej *et al.* (2010), e cada objeto da cena foi chamado para renderização apenas uma vez. Mesmo com o algoritmo de *clipping* ativo, o

overhead de renderizar a cena ainda é pequeno comparado com a execução do *shader* sobre cada fragmento da imagem.

No estudo sobre o número de fatias de renderização, a taxa de *FPS* sofreu pouca alteração para um número suficientemente grande de fatias para ter uma boa renderização. Segundo Oortmerssen (2015), 16 fatias fornece um bom balanço entre desempenho e qualidade visual do panorama renderizado. Valores maiores do que isso causam mudanças imperceptíveis na renderização final (Figura 31), dadas as configurações de *buffer* e detalhamento da cena. Era esperado que aumentar o número de renderizações da cena, multiplicado pelo número de agentes que observam a cena, causasse uma demora significativa a mais além da obtida pela implementação do modelo de Ondřej *et al.* (2010). Porém, como o tamanho do *buffer* foi mantido igual ao original, e cada renderização apenas utiliza uma porção deste *buffer*, a pequena variação no desempenho pode ser atribuída ao *overhead* de trocar as matrizes de projeção e à múltiplas chamadas para renderização do OpenGL, no lugar de apenas uma. Mesmo assim, para cada agente a variação observada foi pequena.

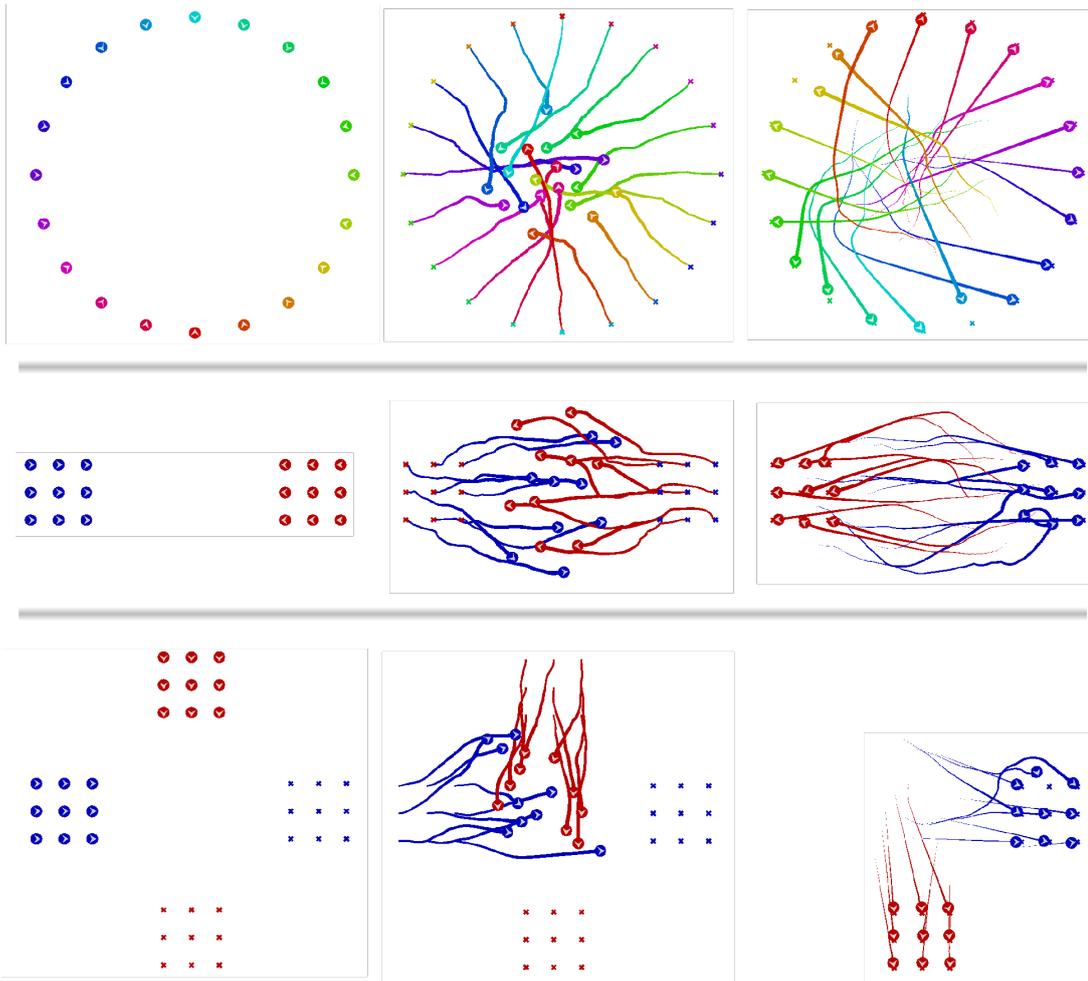
Por fim, foi analisado o número de agentes. Mesmo com todos os outros parâmetros sendo alterados para cada execução, o que foi observado é que há pouca variância para um valor fixo de agentes. Para os exemplos testados, isso significa que a variação dos outros parâmetros foram insignificantes comparada à variação do número de agentes. Para conseguir uma melhoria de desempenho significativa, é necessário otimizar o algoritmo para paralelizar o máximo possível os cálculos realizados por cada agente, pois é onde encontra-se o gargalo da técnica. Para cada agente é realizada uma renderização completa de um *buffer* de visão, com tamanho especificado no trabalho de Ondřej *et al.* (2010). Após a renderização, uma série de cálculos são feitos para determinar os novos valores para a velocidade e orientação do agente. E esses passos são realizados uma vez para cada agente na cena. A medida que o número de agentes na cena aumenta, é observada uma queda de desempenho na execução da técnica.

3.5 Trabalhos futuros

3.5.1 Diferentes regiões de visão

Seres humanos, e muitos outros animais, possuem dois olhos, um fato que possibilita uma visão estereoscópica do mundo, permitindo que exista noção de profundidade entre os objetos que está enxergando. As duas imagens são sobrepostas pelo cérebro de modo a criar

Figura 31 – Diferentes cenários usados nos testes. Na primeira linha, uma cena circular, onde os agentes encontram-se equidistantes do centro, e tem como objetivo o ponto diametralmente oposto. Na segunda linha, o caso onde dois grupos devem se atravessar para atingir seus objetivos, localizados na posição em que os agentes do outro grupo se encontram. Na terceira linha, dois grupos marcham em direções que são perpendiculares entre si. Da esquerda para a direita, as figuras indicam a passagem do tempo. Na primeira está a configuração inicial da cena, e na segunda está um momento em que há um grande número de agentes desviando dos demais para conseguirem atingir seu objetivo, e na terceira, o momento que os agentes chegam ao seu destino.



Fonte: Autor.

uma única imagem. Mas esse campo de visão não é uniforme. Diferentes regiões do campo de visão atuam de maneira diferente na locomoção das pessoas. Enquanto há um foco na região central, onde a imagem é formada pela captura dos dois olhos simultaneamente, nas regiões periféricas a imagem é borrada e capturada por apenas um dos olhos. O cérebro processa essas informações e reage de maneira diferente para cada uma. Enquanto o foco da visão está, em geral, direcionado para onde o indivíduo está se locomovendo, sua visão periférica é capaz de detectar movimentos bruscos, que, com o devido reflexo, faz com que o pedestre seja capaz de

desviar de um obstáculo inesperado.

Nos modelos atuais de visão sintética, essa característica do campo de visão não existe. Todos os fragmentos da imagem capturada produzem o mesmo efeito sobre a decisão tomada pelo agente virtual. Usando a técnica de visão panorâmica por fatias, é possível atribuir diferentes funções de controle do agente para cada região da visão, e a criação de regiões heterogêneas de visão, com ângulos arbitrários podem ser configuradas. Um possível avanço como resultado dessa técnica seria a possibilidade da criação de um modelo capaz de representar com maior fidelidade o comportamento do personagem em relação ao seu sentido da visão.

Seres humanos também são capazes de rotacionar suas cabeças em torno do pescoço, e fazem isso com frequência quando estão se locomovendo, para garantir que nenhuma ameaça irá cruzar seu caminho. Dependendo do nível de atenção, a frequência em que isso ocorre pode variar. Uma possibilidade é considerar, estatisticamente, qual a porcentagem de tempo dedicada a cada região visualizada e distribuir a informação capturada de forma ponderada para o controle do personagem.

Além disso, com a flexibilidade adquirida pela técnica, seria possível simular diferentes campos de visão de diferentes espécies de avatares, representando diferentes posicionamentos para sensores ópticos nos seres vivos. Entretanto, seria necessário mudar também a modelagem das funções de controle, já que as funções criadas nos trabalhos existentes foram inspiradas no comportamento humano observado.

Figura 32 – Comparando diferentes espécies de agentes e seus respectivos campos de visão. Do lado esquerdo, um ser humano sendo visto por cima. Do lado direito, um peixinho dourado. A região azul corresponde ao campo de visão do olho esquerdo. A região vermelha a do olho direito. A região de cor roxa indica onde há sobreposição das imagens dos dois olhos, composta no cérebro do indivíduo. A região roxa é denominada de área de visão binocular, ou estereoscópica, e as demais regiões são áreas de visão monocular, ou visão periférica. Os ângulos indicados na imagem não correspondem às medidas reais, sendo apenas ilustrativos das diferentes regiões.



Fonte: Autor

3.5.2 *Diversificação dos agentes*

Em um ambiente real, pedestres tendem a seguir padrões globais de comportamento semelhantes, dependendo da situação. Mesmo assim, é importante que haja variabilidade entre os agentes. Uma multidão real não é composta por indivíduos idênticos, e sim por vários exemplares com diferentes características. Cada agente pode se diferenciar dos demais em tamanho do campo de visão, altura, largura, velocidade, qualidade da visão, diferentes comportamentos e até mesmo deficiências que possam alterar sua locomoção. É interessante realizar esse estudo atribuindo para cada agente valores diferentes para que cada um seja único. Espera-se que os comportamentos globais sejam mantidos, mas que ao observar a multidão com maior proximidade, seja possível perceber diferentes comportamentos entre os agentes, dando uma sensação de maior naturalidade da simulação.

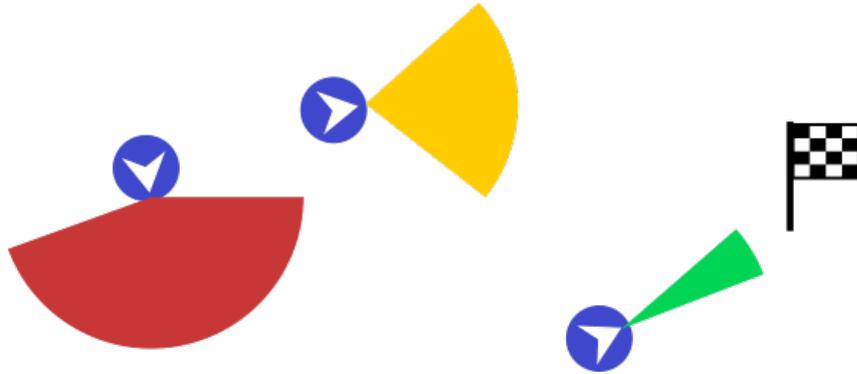
3.5.3 *Relaxamento da restrição do objetivo*

Uma das suspeitas de por que o agente colide com os outros, ou entra em situações onde não há saída, é por causa da restrição do objetivo. O custo para o agente se afastar de seu objetivo é muito elevado, fazendo com que ele dê preferência a entrar em cenários indesejáveis. Uma forma que foi pensada para solucionar esse problema é de relaxar o custo do objetivo em função da distância. Isso quer dizer que, quanto mais longe o agente está do seu objetivo, menos importa para ele seguir uma direção exata. O objetivo agora seria tratado como um *range*, e não apenas como um vetor em direção a um ponto, permitindo o agente ter mais liberdade para desviar dos obstáculos, mas se manter na direção de seu objetivo.

3.5.4 *Movimentação em três dimensões*

Os modelos de simulação de multidão baseados em visão sintética existentes consideram que toda a locomoção possível acontece em duas dimensões. Por isso, o agente só é capaz de desviar de seus obstáculos rotacionando em um único eixo, e caso isso não resolva o problema, é obrigado a diminuir sua velocidade. Adicionar a dimensão vertical no modelo poderia adicionar novas características à simulação, como possibilidade dos agentes se locomoverem em três dimensões, como em terrenos com relevo, simular agentes aéreos ou aquáticos, que podem se locomover no fluido em três dimensões, ou até mesmo possibilitar que o agente mude sua animação de locomoção para desviar de algum obstáculo na vertical, como um pedestre alto

Figura 33 – O agente tolera ângulos em relação ao seu objetivo. À medida que se aproxima de seu objetivo, o agente deveria tolerar menos desvios de sua trajetória. Nesse caso, o agente tem que manter a direção do objetivo dentro do ângulo de tolerância a sua frente. Na imagem, o agente com campo de tolerância verde permite uma variação menor do que o agente que possui o campo vermelho por estar mais próximo do objetivo.



Fonte: Autor

tentando passar por baixo de um tronco de árvore.

3.6 Considerações finais

Neste capítulo foi apresentada uma técnica para expandir a capacidade visual dos agentes virtuais portadores de visão sintética de modo que eles possam enxergar em ângulos arbitrários de até 360° . O resultado mostrou-se positivo, indicando que a modificação foi capaz de introduzir novos comportamentos aos agentes, possibilitando a simulação de pedestres com diferentes capacidades visuais, ou até mesmo agentes de outras espécies. Além disso, a perda de desempenho causada pela adição da nova técnica foi bem menor que inicialmente esperado. Nos dois modelos em que a técnica foi aplicada, o número de agentes mostrou-se o maior responsável pela perda de desempenho. No próximo capítulo, é discutida uma forma de introduzir o conceito de percepção de densidade de multidões no modelo GBM.

4 DENSIDADE

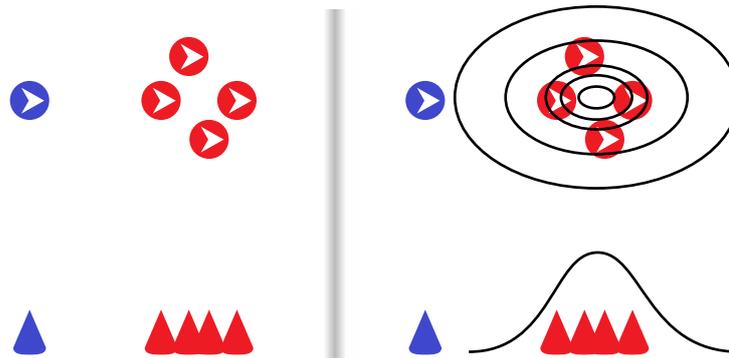
4.1 Percepção de densidade

Como apontado em Best *et al.* (2014), a percepção de densidade é um fator chave na locomoção de agentes virtuais, mostrando que há uma relação entre a densidade e a dinâmica dos pedestres. A medida que a densidade aumenta, a velocidade em que eles se movem tende a diminuir. No mesmo trabalho, os autores modelam os comportamentos de uma multidão de agentes governados pela densidade sobre um modelo de simulação de multidões baseado em velocidade. O modelo detalhado na Seção 2.3, daqui em diante denominado *HSV (Hughes' Synthetic Vision)*, é uma expansão sobre o trabalho de Ondřej *et al.* (2010), o *OSV*, acrescentando as noções de percepção de densidade propostas por Best *et al.* (2014).

A densidade de uma multidão é um valor que indica a distribuição de personagens em uma determinada área, geralmente medida em número de pedestres por metro quadrado. O modelo de Best *et al.* (2014), de agora em diante denominado *BM (Best's Model)*, calcula o valor da densidade da multidão para cada agente considerando uma área elíptica centrada em um ponto a sua frente. Isso é feito de modo que a densidade seja calculada no ponto para onde o agente azul está se deslocando. Essa direção pode ser modificada variando um ângulo θ para que seja possível encontrar uma nova direção na qual o personagem deva se movimentar a fim de minimizar a densidade à sua frente. A Figura 35 ilustra como é feito o cálculo da densidade do agente azul considerando um ângulo θ igual a zero. A contribuição para o fator de densidade de cada agente vermelho é o seu valor sobre a curva gaussiana, que é somada para gerar o valor total. No modelo HSV, a densidade é calculada da mesma maneira. Porém, seu valor é aplicado de maneira diferente no algoritmo de controle do agente virtual.

Um detalhe importante sobre a forma que a aquisição dessa informação foi proposta para o modelo HSV é que ela é obtida antes do passo de captura da imagem do agente virtual, considerando o posicionamento de todos os agentes que estão ao redor de um ponto à frente do agente. Isso significa que a informação de densidade da multidão não chega ao personagem através do seu fluxo óptico, fazendo com que o modelo não seja puramente baseado em visão sintética, mas sim um modelo híbrido. Em uma situação real, um indivíduo em uma multidão não é capaz de enxergar através de obstáculos devido a propriedade de oclusão da luz. Esse personagem então não deveria ser capaz de sentir a influência de um outro indivíduo pode levar a resultados menos coerentes com a realidade. Além disso, para cada passo da simulação,

Figura 34 – Como é feito o cálculo da densidade. O agente azul é o observador, e os demais são obstáculos. Na parte superior, é mostrada a cena vista do alto, sobre a cabeça dos agentes. Na parte inferior, é mostrada uma visão lateral da mesma cena. Na esquerda é ilustrada a configuração da cena em um determinado estado, e na direita, o gráfico da função que indica a contribuição para o fator de densidade de cada obstáculo é sobreposto. O gráfico descreve uma função gaussiana cujo parâmetro é apenas a distância, onde a contribuição para o fator da densidade é maior ao longo do eixo da velocidade do agente do que em relação ao eixo perpendicular, o que significa que obstáculos aglomerados na frente do agente representam um risco maior de colisão. Visto de cima, a equação para cada posição parece elíptica por causa da prioridade que se dá para o afastamento no eixo perpendicular ao movimento do agente.



Fonte: Autor.

apenas um valor de densidade é calculado para cada agente, sendo passado como entrada de seu estado. Esse valor não é usado para encontrar o ângulo θ que minimiza o valor da densidade, mas sim para modificar a função de limiar que define quando o agente deve reagir a um obstáculo. O modelo HSV também continua com as mesmas limitações do modelo OSV, como por exemplo, o fato de não utilizar toda a informação obtida pelo sistema visual do agente. No lugar, o personagem considera apenas o fragmento da imagem que representa os valores mais significativos de ameaça de colisão, e toma uma decisão baseada neles, perdendo assim a capacidade de prever colisões que possam ser causadas como consequência de sua escolha. O modelo proposto em Dutra *et al.* (2015) resolve alguns desses problemas, apesar de não contemplar a percepção de densidade dos obstáculos.

4.2 Técnica proposta

Nessa Sessão, é proposta uma maneira para expandir o modelo GBM de modo que seja possível capturar a informação da densidade a partir da imagem da visão do agente virtual, considerando as informações contidas em seu estado a cada passo da simulação, e usando seu valor para computar o resultado uma função de custo. Minimizando essa função de custo, é

possível descobrir uma nova velocidade e orientação que diminuam a densidade de obstáculos à frente do observador. Para que isso aconteça, é necessário que essa função tenha custo mínimo quando não houver nenhum obstáculo à frente do personagem, e custo bem elevado quando um grande número de obstáculos estiverem em seu caminho. Portanto, dada uma velocidade v e um ângulo θ , é definido um novo ponto onde o agente estará no próximo passo da simulação, e um valor de densidade pode ser calculado para aquele ponto:

$$C_d(v_o, \theta_o) = \sum_{i \in A} \frac{e^{-\frac{0.5}{\sigma^2} \|d_{o|i}^f\|^2}}{\sqrt{2\pi\sigma}}. \quad (4.1)$$

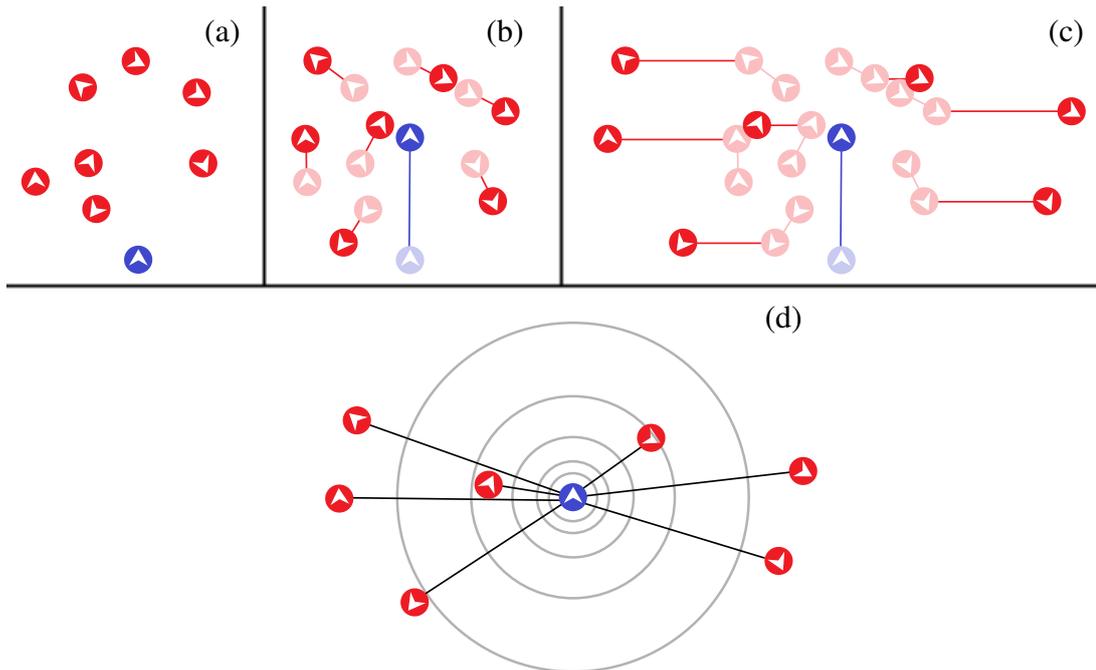
O modelo GBM utiliza duas funções de custo para gerar o custo total. A primeira função, C_m , indica o custo da movimentação do personagem, com o objetivo de possibilitar a detecção de colisão com os obstáculos da cena. Essa função possuirá valor elevado quando houver a possibilidade de colisão e ela for imediata, e possuirá um valor baixo quando não houver risco. A segunda função, C_o , representa o custo do agente se afastar do seu objetivo. Quando o agente está indo rumo ao seu objetivo, a função tem custo mínimo. À medida que o ângulo entre a velocidade do personagem e o seu objetivo aumenta, o custo também aumenta. Essas duas funções são somadas ao final, gerando a função de custo total C_t . Uma vez calculada a função de custo, o algoritmo descobre a direção na qual os parâmetros v_o e θ_o devem variar para diminuir o custo, usando o gradiente ∇C_t (Apêndice B).

Como essa técnica é uma adição da percepção de densidade sobre o modelo GBM, será chamada de agora em diante de GBM-D.

4.3 Implementação

Para cada agente o , seu valor na função gaussiana descrita na Equação 4.1 é calculado baseado na distância relativa $d_{o|i}$ entre o agente observador o e o obstáculo i . Para aumentar a antecipação do algoritmo, o ponto onde o obstáculo se encontra pode ser considerado extrapolando sua trajetória ao longo do mesmo período de tempo em que foi considerado o deslocamento do observador. Best *et al.* (2014) também descrevem um deslocamento da posição dos agentes observados em um eixo perpendicular à velocidade do observador. Esse passo é equivalente a considerar que a função gaussiana é elíptica, uma vez que os agentes estarão proporcionalmente espalhados ao longo da direção em que a contribuição para a densidade deve ser menor, e se manterão em suas posições ao longo da direção da velocidade.

Figura 35 – Cálculo do fator de densidade. Na figura (a), vemos o estado atual da cena. Nosso cálculo será referente à visão do agente azul. Em (b), extrapolamos a posição em que os personagens estarão após um *timestep*, deslocando os agentes na direção de suas velocidades. Na figura (c), um deslocamento proporcional à distância de cada agente para o eixo da velocidade do observador é aplicado, para diminuir a relevância para o cálculo da densidade de objetos que não estejam na trajetória do personagem azul. Em (d), vemos as posições finais após todas as transformações. Essas distâncias entre os agentes vermelhos e o agente azul serão utilizadas para o cálculo do fator de densidade.



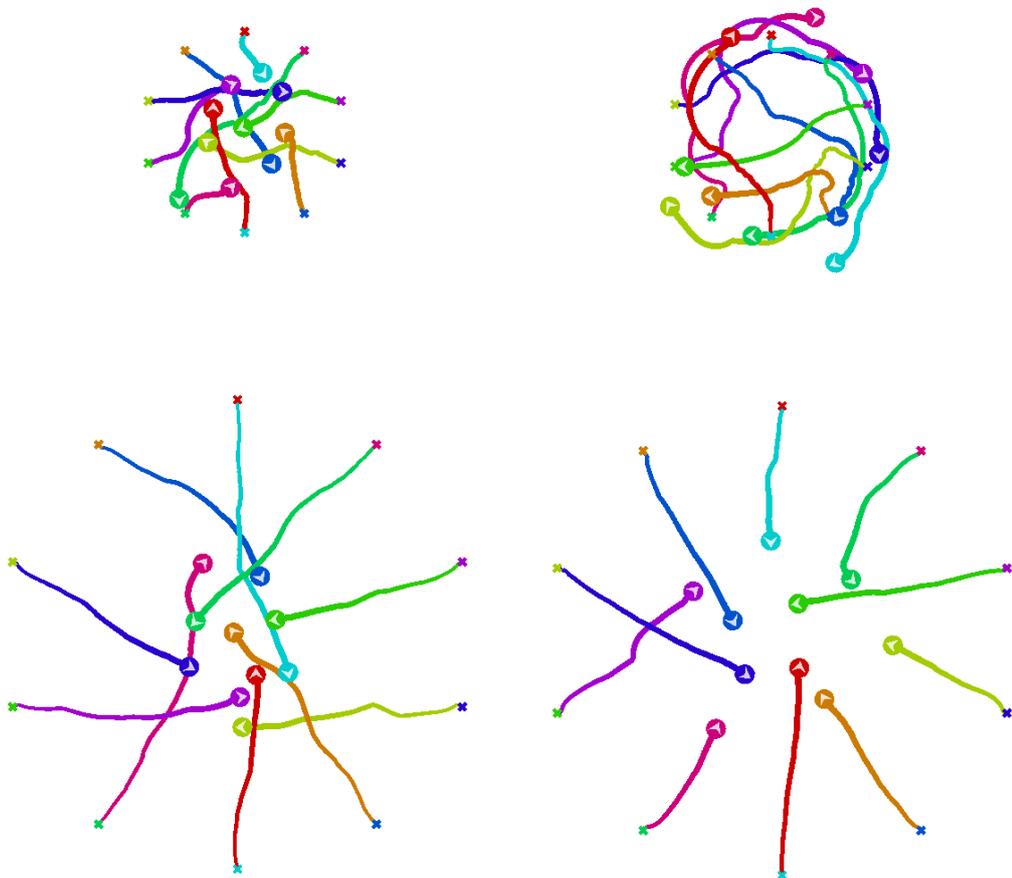
Fonte: Autor.

Diferentemente da técnica proposta por Hughes *et al.* (2015), como a função é calculada usando a visão do agente, o valor precisa ser computado dentro do *shader*, junto com a função de custo de movimentação C_m . Cada fragmento que representa um objeto possuirá um estado representando sua dinâmica, e será considerado como um obstáculo único, dando sua contribuição para a função de custo total da mesma maneira que o algoritmo de Dutra *et al.* (2015) faz com respeito à movimentação. Isso implica que somente os objetos que estiverem dentro do campo de visão do personagem, e não estiverem sendo oclusos por outros obstáculos serão considerados para fins de orientação do observador.

4.4 Resultados

No primeiro teste realizado, os agentes foram colocados ao redor de um círculo. O objetivo é estudar como esses agentes se comportam ao passar pelo meio.

Figura 36 – Um exemplo de testes comparativos, usando os valores $\sigma_1 = 1$ e $\sigma_2 = 0.2$. O cenário circular com 8 agentes foi utilizado. Nas imagens superiores, o cenário de círculo com raio pequeno foi utilizado, enquanto nas imagens inferiores, um círculo com maior raio foi utilizado. Na esquerda, encontra-se o modelo GBM, enquanto na direita está o GBM-D. É possível ver que, em situações de alta densidade, para os parâmetros utilizados, houve uma dispersão muito grande entre os agentes, levando a um comportamento indesejado. Porém, no caso do círculo maior, em que há maior espaço para os agentes percorrerem, os agentes se mantiveram mais espalhados, como previsto.

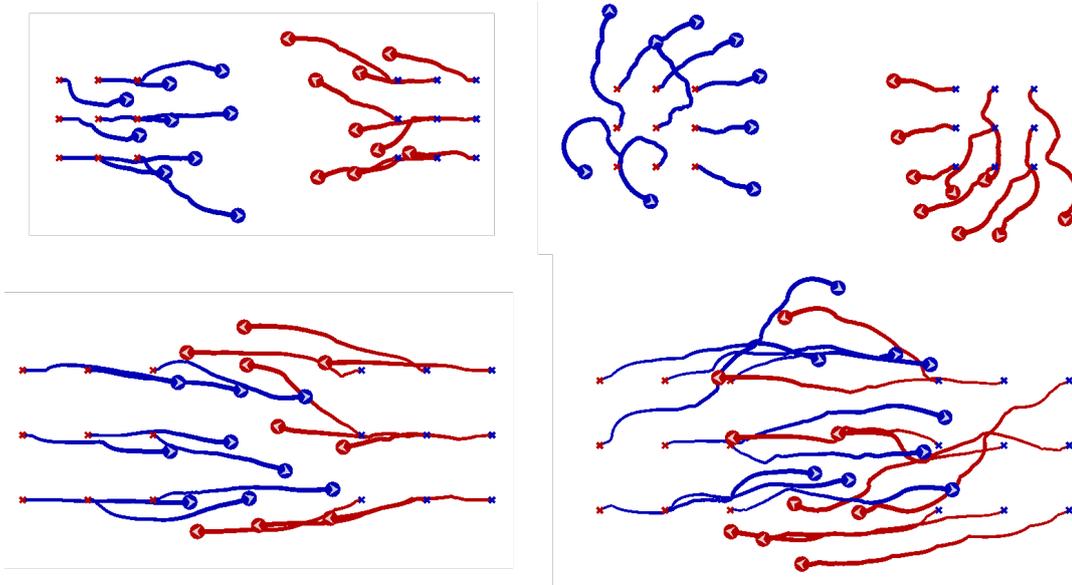


Fonte: Autor.

Para a configuração de parâmetros escolhida, foi possível criar o comportamento desejado, em que os agentes passam com maior distância pela região central de maior aglomeração. No caso em que os agentes se encontram mais próximos, a função de custo ficou com valor tão elevado que afastou os agentes da trajetória mais curta, fazendo com que eles dessem a volta por fora do círculo. Quanto menor o valor de σ_1 , mais parecido o GBM-D fica com o algoritmo GBM, enquanto o parâmetro σ_2 define um intervalo de densidades no qual o agente deve reagir. É possível, então, calibrar um balanço entre as duas técnicas modificando esses parâmetros.

Em seguida foi realizado um teste no cenário de troca de grupos, onde dois blocos de agentes devem se atravessar para chegar na posição em que o outro grupo se encontrava inicialmente.

Figura 37 – Outro exemplo, usando a cena onde dois grupos devem trocar de lugar. A estrutura segue a mesma ordem da Figura 36

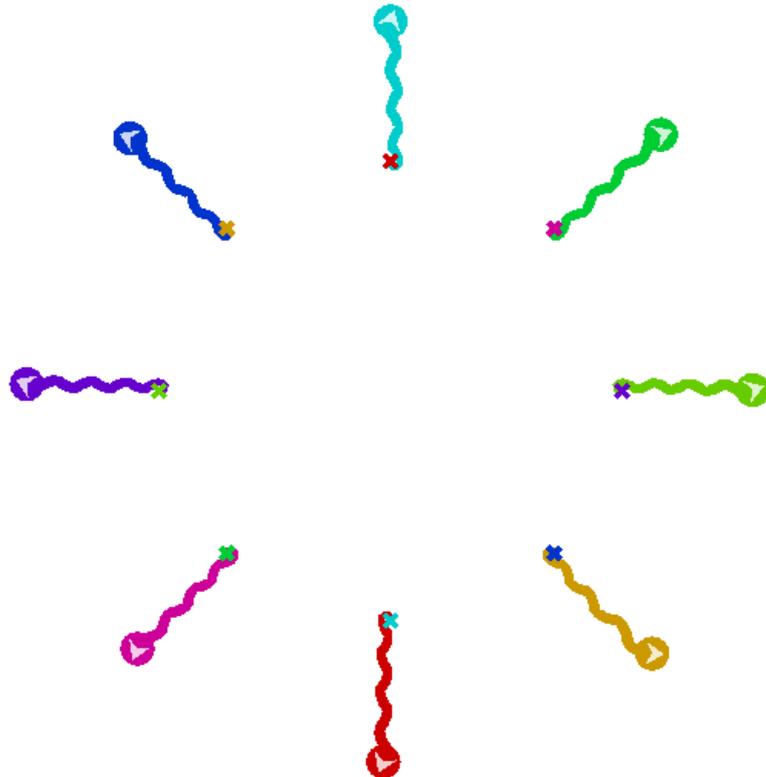


Fonte: Autor.

Por conta dos altos valores dos parâmetros utilizados, os agentes criaram uma tendência de se afastar do seu próprio grupo, mesmo que este esteja se movendo na mesma direção que ele, pois se afastar do grupo diminui a densidade percebida pelo agente.

Apesar de ter trazido notável melhoria para o modelo OSV, quando testado no GBM houve casos em que aconteceu uma melhoria, e outros casos em que os agentes tomaram rotas indesejadas. Entretanto, o modelo GBM já utiliza todo seu campo de visão, dando uma capacidade de antecipação e noção de consequência de suas atitudes. Então o modelo GBM, mesmo que de forma bem menos intensa, já evita, de alguma forma, aglomerações mesmo sem a noção de densidade sendo passada de maneira explícita para o algoritmo. A função de custo de densidade permitiu uma configuração da sensibilidade da multidão a essas regiões de densidade.

Figura 38 – Caso o parâmetro σ_1 seja muito alto, a função de custo da densidade irá predominar sobre as demais funções de custo, e os agentes jamais irão evitar colisões ou ir em direção ao seu objetivo, pois quanto mais se afastarem, menor será a densidade. O experimento exagerado mostra que a função de custo de fato funciona, mesmo que criando um resultado indesejado. É possível ver um padrão de zigue zague na imagem, pois quando não consegue ver nenhum outro agente, ele começa a seguir a função de custo do objetivo C_o , mas retoma seu afastamento dos demais quando algum agente entra em seu campo de visão.



Fonte: Autor.

4.5 Trabalhos futuros

4.5.1 *Buscar novos cenários*

Uma hipótese para a técnica não ter obtido os resultados esperados é que foram testados cenários onde o modelo GBM já foi validado e demonstrou contornar bem as adversidades. Novos cenários onde o modelo GBM tem dificuldades de locomoção, ou não se comporta da maneira esperada devem ser pensados e testados.

4.5.2 Comparação com HSV

Um estudo que não foi executado por restrições de tempo é a comparação do modelo GBM-D com o modelo HSV. É interessante saber se há ganho do uso do modelo GBM em relação ao HSV, e se adicionar a percepção de densidade no modelo adiciona alguma vantagem na comparação.

4.6 Considerações finais

Neste trabalho foi proposta um novo modelo de simulação de multidões com uso de visão sintética, o GBM-D. A inserção da percepção de densidade trouxe alguma diferença para o comportamento dos agentes. Alguns comportamentos indesejados, e outros que podem ser úteis. A técnica precisa ser testada em mais casos diferentes, com uma gama maior de parâmetros para demonstrar sua eficácia.

REFERÊNCIAS

- ALI, S.; NISHINO, K.; MANOCHA, D.; SHAH, M. In: **Modeling, Simulation and Visual Analysis of Crowds: A Multidisciplinary Perspective**. [S.l.]: Springer New York, 2013. ISBN 978-1-4614-8482-0.
- BERG, J. van den; GUY, S.; LIN, M.; MANOCHA, D. Reciprocal n-body collision avoidance. In: **Robotics Research**. [S.l.]: Springer, 2011. v. 70, p. 3–19.
- BERG, J. van den; LIN, M.; MANOCHA, D. Reciprocal velocity obstacles for real-time multi-agent navigation. In: **IEEE Int. Conf. on Robotics and Automation**. [S.l.: s.n.], 2008. p. 1928–1935. ISSN 1050-4729.
- BEST, A.; NARANG, S.; CURTIS, S.; MANOCHA, D. Densesense: Interactive crowd simulation using density-dependent filters. In: **Symposium on Computer Animation**. [S.l.: s.n.], 2014. p. 97–102.
- BRAUN, A.; MUSSE, S. R.; OLIVEIRA, L. P. L. de; BODMANN, B. E. J. Modeling individual behaviors in crowd simulation. **16th International Conference on Computer Animation and Social Agents**, p. 143 – 148, 2003. ISSN 1087-4844.
- BURSTEDDE, C.; KLAUCK, K.; SCHADSCHNEIDER, A.; ZITTARTZ, J. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. **Physica A Statistical Mechanics and its Applications**, v. 295, p. 507–525, jun. 2001.
- CHARALAMBOUS, P.; CHRYSANTHOU, Y. The pag crowd: A graph based approach for efficient data-driven crowd simulation. **Comput. Graph. Forum**, v. 8, n. 33, p. 95–108, 2014.
- CHENNEY, S. Flow tiles. In: **Proc. of the 2004 ACM SIGGRAPH/Eurographics symp. on Computer animation**. [S.l.: s.n.], 2004. p. 233–242. ISBN 3-905673-14-2.
- COURTY, N.; MUSSE, S. “Simulation of large crowds in emergency situations including gaseous phenomena”. In: **Computer Graphics International 2005**. [S.l.: s.n.], 2005. p. 206 – 212. ISSN 1530-1052.
- CUTTING, J.; VISHTON, P.; BRAREN, P. How we avoid collisions with stationary and moving objects. **Psychological Review**, US: American Psychological Association, v. 102, n. 4, p. 627–651, 1995. ISSN 1939-1471.
- DIJKSTRA, J.; TIMMERMANS, H. J. P.; JESSURUN, A. J. A multi-agent cellular automata system for visualising simulated pedestrian activity. In: **in S. Bandini and T. Worsch (Eds.), Theoretical and Practical Issues on Cellular Automata - Proceedings on the 4th International Conference on Cellular Automata for research and Industry**. [S.l.]: Springer Verlag, 2000. p. 29–36.
- DUTRA, T.; CAVALCANTE-NETO, J.; VIDAL, C. Gradient-based steering for vision-based crowd simulation algorithms. 2015.
- DUTRA, T.; CAVALCANTE-NETO, J.; VIDAL, C.; MUSSE, S. A multipotential field model for crowds with scalable behaviors. **Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI - Conference on**, p. 31–38, 2013.

FIORINI, P.; SHILLER, Z. Motion planning in dynamic environments using velocity obstacles. **The International Journal of Robotics Research**, SAGE Publications, v. 17, n. 7, p. 760–772, 1998.

GIBSON, J. J. Visually controlled locomotion and visual orientation in animals. **British Journal of Psychology, Blackwell Publishing Ltd**, v. 49, n. 3, p. 182–194, 1958. ISSN 2044-8295.

GOLAS, A.; NARAIN, R.; CURTIS, S.; LIN, M. Hybrid long-range collision avoidance for crowd simulation. **Visualization and Computer Graphics, IEEE Transactions on**, v. 20, n. 7, p. 1022–1034, July 2014. ISSN 1077-2626.

GUY, S. J.; CHHUGANI, J.; KIM, C.; SATISH, N.; LIN, M.; MANOCHA, D.; DUBEY, P. Clearpath: Highly parallel collision avoidance for multi-agent simulation. In: **Proc. of the 2009 ACM SIGGRAPH/Eurographics Symp. on Computer Animation**. New York, NY, USA: ACM, 2009. (SCA '09), p. 177–187. ISBN 978-1-60558-610-6. Disponível em: <<http://doi.acm.org/10.1145/1599470.1599494>>.

GUY, S. J.; KIM, S.; LIN, M. C.; MANOCHA, D. Simulating heterogeneous crowd behaviors using personality trait theory. In: **Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation**. New York, NY, USA: ACM, 2011. (SCA '11), p. 43–52. ISBN 978-1-4503-0923-3.

HEIGEAS, L.; LUCIANI, A.; THOLLOT, J.; CASTAGNÉ, N. A Physically-Based Particle Model of Emergent Crowd Behaviors. In: **Graphicon**. [S.l.: s.n.], 2003.

HELBING, D.; FARKAS, I.; VICSEK, T. Simulating dynamical features of escape panic. **Nature**, [London: Macmillan Journals], 1869-, v. 407, n. 6803, p. 487–490, 2000.

HELBING, D.; MOLNÁR, P. Social force model for pedestrian dynamics. **Physical Review E**, American Physical Society, v. 51, n. 5, p. 4282–4286, 1995.

HUGHES, R.; ONDŘEJ, J.; DINGLIANA, J. Davis : Density-adaptive synthetic-vision based steering for virtual crowds. **ACM SIGGRAPH Conference on Motion in Games 2015, Paris, France**, p. 79 – 84, 2015.

HUGHES, R. L. “A continuum theory for the flow of pedestrians”. **Transportation Research Part B: Methodological**, v. 36, n. 6, p. 507 – 535, 2002. ISSN 0191-2615.

HUGHES, R. L. “The flow of human crowds”. **Annual Review of Fluid Mechanics**, v. 35, n. 1, p. 169–182, 2003.

JIANG, H.; XU, W.; MAO, T.; LI, C.; XIA, S.; WANG, Z. “Continuum crowd simulation in complex environments”. **Computers & Graphics**, v. 34, n. 5, p. 537 – 544, 2010. ISSN 0097-8493.

JU, E.; CHOI, M.; PARK, M.; LEE, J.; LEE, K.; TAKAHASHI, S. Morphable crowds. **ACM Trans. Graph.**, ACM, New York, NY, USA, v. 29, p. 140:1–140:10, 2010. ISSN 0730-0301.

KAPADIA, M.; BADLER, N. I. Navigation and steering for autonomous virtual humans. **Wiley Interdisciplinary Reviews: Cognitive Science, John Wiley & Sons, Inc.**, v. 4, n. 3, p. 263–272, 2013. ISSN 1939-5086.

KAPADIA, M.; SINGH, S.; HEWLETT, W.; FALOUTSOS, P. Egocentric affordance fields in pedestrian steering. In: **Proceedings of the 2009 symposium on Interactive 3D graphics and games**. New York, NY, USA: ACM, 2009. (I3D '09), p. 215–223. ISBN 978-1-60558-429-4.

KARAMOUZAS, I.; HEIL, P.; BEEK, P. van; OVERMARS, M. H. A predictive collision avoidance model for pedestrian simulation. In: **Motion in Games**. [S.l.]: Springer, 2009, (LNSC, v. 5884). p. 41–52. ISBN 978-3-642-10346-9.

KHATIB, O. Real-time obstacle avoidance for manipulators and mobile robots. In: **Robotics and Automation. Proceedings. 1985 IEEE International Conference on**. [S.l.: s.n.], 1985. v. 2, p. 500–505.

KIRCHNER, A.; NAMAZI, A.; NISHINARI, K.; SCHADSCHNEIDER, A. Role of conflicts in the floor field cellular automaton model for pedestrian dynamics. In: **2nd International Conference on Pedestrians and Evacuation Dynamics**. [S.l.: s.n.], 2003. p. 51–62.

KIRCHNER, A.; SCHADSCHNEIDER, A. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. **Physica A: Statistical Mechanics and its Applications**, v. 312, n. 1–2, p. 260 – 276, 2002. ISSN 0378-4371.

LAMARCHE, F.; DONIKIAN, S. Crowd of virtual humans: a new approach for real time navigation in complex and structured environments. **Computer Graphics Forum**, Blackwell Publishing, Inc, v. 23, n. 3, p. 509–518, 2004. ISSN 1467-8659. Disponível em: <<http://dx.doi.org/10.1111/j.1467-8659.2004.00782.x>>.

LEE, K. H.; CHOI, M. G.; HONG, Q.; LEE, J. Group behavior from video: a data-driven approach to crowd simulation. In: **Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation**. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007. (SCA '07), p. 109–118. ISBN 978-1-59593-624-0.

LERNER, A.; CHRYSANTHOU, Y.; LISCHINSKI, D. Crowds by example. **Computer Graphics Forum**, Wiley Online Library, v. 26, n. 3, p. 655–664, 2007.

LERNER, A.; FITUSI, E.; CHRYSANTHOU, Y.; COHEN-OR, D. Fitting behaviors to pedestrian simulations. In: **Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation**. New York, NY, USA: ACM, 2009. (SCA '09), p. 199–208. ISBN 978-1-60558-610-6.

LOSCOS, C.; MARCHAL, D.; MEYER, A. Intuitive crowd behavior in dense urban environments using local laws. In: **Theory and Practice of Computer Graphics, 2003. Proceedings**. [S.l.: s.n.], 2003. p. 122–129.

MARTIN, G. Visual fields in woodcocks *scolopax rusticola* (scolopacidae; charadriiformes). **Journal of Comparative Physiology A**, Springer-Verlag, v. 174, n. 6, p. 787–793, 1994. ISSN 0340-7594. Disponível em: <<http://dx.doi.org/10.1007/BF00192728>>.

METOYER, R. A.; HODGINS, J. K. Reactive pedestrian path following from examples. **The Visual Computer**, Springer-Verlag, v. 20, n. 10, p. 635–649, 2004. ISSN 0178-2789.

MOUSSAÏD, M.; HELBING, D.; THERAULAZ, G. How simple rules determine pedestrian behavior and crowd disasters. **Proceedings of the National Academy of Sciences**, v. 108, n. 17, p. 6884–6888, 2011. Disponível em: <<http://www.pnas.org/content/108/17/6884.abstract>>.

MOUSSAÏD, M.; PEROZO, N.; GARNIER, S.; HELBING, D.; THERAULAZ, G. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. **PLoS ONE**, Public Library of Science, v. 5, n. 4, p. e10047, April 2010.

MUSSE, S. R. **Human crowd modelling with various levels of behaviour control**. Tese (Doutorado), Lausanne, 2000. Disponível em: <<http://library.epfl.ch/theses/?nr=2128>>.

MUSSE, S. R.; THALMANN, D. Hierarchical model for real time simulation of virtual human crowds. **IEEE Transactions on Visualization and Computer Graphics**, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 7, p. 152–164, April 2001. ISSN 1077-2626.

NARAIN, R.; GOLAS, A.; CURTIS, S.; LIN, M. C. Aggregate dynamics for dense crowd simulation. **ACM Transactions on Graphics**, ACM, New York, NY, USA, v. 28, p. 122:1–122:8, 2009. ISSN 0730-0301.

NOSER, H.; RENAULT, O.; THALMANN, D.; THALMANN, N. M. Navigation for digital actors based on synthetic vision, memory, and learning. **Computers & Graphics**, v. 19, n. 1, p. 7 – 19, 1995. ISSN 0097-8493.

ONDŘEJ, J.; PETTRÉ, J.; OLIVIER, A.-H.; DONIKIAN, S. A synthetic-vision based steering approach for crowd simulation. **ACM Trans. Graph.**, ACM, New York, NY, USA, v. 29, n. 4, p. 123:1–123:9, jul. 2010. ISSN 0730-0301.

OORTMERSSEN, W. van. **PanQuake**. 2015. [Http://strlen.com/gfxengine/panquake/](http://strlen.com/gfxengine/panquake/).

PARAVISI, M.; WERHLI, A.; JUNIOR, J. J.; RODRIGUES, R.; JUNG, C. R.; MUSSE, S. R. Continuum crowds with local control. **Computer Graphics International**, p. 108–115, 2008.

PARIS, S.; PETTRÉ, J.; DONIKIAN, S. Pedestrian reactive navigation for crowd simulation: a predictive approach. **Computer Graphics Forum**, Blackwell Publishing Ltd, v. 26, n. 3, p. 665–674, 2007. ISSN 1467-8659. Disponível em: <<http://dx.doi.org/10.1111/j.1467-8659.2007.01090.x>>.

PATIL, S.; BERG, J. van den; CURTIS, S.; LIN, M. C.; MANOCHA, D. Directing crowd simulations using navigation fields. **IEEE Transactions on Visualization and Computer Graphics**, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 17, p. 244–254, February 2011. ISSN 1077-2626.

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**. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007. (SCA '07), p. 99–108. ISBN 978-1-59593-624-0. Disponível em: <<http://dl.acm.org/citation.cfm?id=1272690.1272705>>.

PELECHANO, N.; ALLBECK, J. M.; BADLER, N. I. Virtual crowds: Methods, simulation, and control. **Synthesis Lectures on Computer Graphics and Animation**, Morgan & Claypool Publishers, v. 3, n. 1, p. 1–176, 2008.

PETERS, C.; O’SULLIVAN, C. Bottom-up visual attention for virtual human animation. In: **Computer Animation and Social Agents, 2003. 16th International Conference on**. [S.l.: s.n.], 2003. p. 111–117. ISSN 1087-4844.

- PETTRÉ, 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**. [S.l.: s.n.], 2005. v. 43, n. 44, p. 194.
- PETTRÉ, J.; ONDRÉJ, J.; OLIVIER, A.-H.; CRETUAL, A.; DONIKIAN, S. Experiment-based modeling, simulation and validation of interactions between virtual walkers. In: **Proc. of the 2009 ACM SIGGRAPH/Eurographics Symp. on Computer Animation**. New York, NY, USA: ACM, 2009. (SCA '09), p. 189–198. ISBN 978-1-60558-610-6. Disponível em: <<http://doi.acm.org/10.1145/1599470.1599495>>.
- RENAULT, O.; MAGNENAT-THALMANN, N.; THALMANN, D. A vision-based approach to behavioural animation. **Journal of Visualization and Computer Animation**, John Wiley & Sons, Ltd, v. 1, n. 1, p. 18–21, 1990. ISSN 1099-1778.
- REYNOLDS, C. Flocks, herds and schools: a distributed behavioral model. **SIGGRAPH Comp. Graphics**, ACM, New York, NY, USA, v. 21, n. 4, p. 25–34, 1987. ISSN 0097-8930.
- REYNOLDS, C. Steering behaviors for autonomous characters. In: **Game Developers Conference 1999**. [S.l.: s.n.], 1999. p. 763–782.
- REYNOLDS, C. W. “Interaction with groups of autonomous characters”. In: **Game Developers Conference 2000**. [S.l.: s.n.], 2000. p. 449–460.
- REYNOLDS, C. W. “Big fast crowds on ps3”. In: **Sandbox '06: Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames**. New York, NY, USA: ACM, 2006. p. 113–121. ISBN 1-59593-386-7.
- RIO, K. W.; RHEA, C. K.; WARREN, W. H. Follow the leader: Visual control of speed in pedestrian following. **Journal of Vision**, v. 14, n. 2, 2014.
- SCHADSCHNEIDER, A. Cellular automaton approach to pedestrian dynamics-theory. **Pedestrian and evacuation dynamics**, Springer Verlag, p. 75–85, 2001.
- SHAO, W.; TERZOPOULOS, D. Autonomous pedestrians. **Graphical Models**, v. 69, n. 5-6, p. 246–274, 2007. ISSN 1524-0703.
- SHIMODA, S.; KURODA, Y.; IAGNEMMA, K. Potential field navigation of high speed unmanned ground vehicles on uneven terrain. In: **Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on**. [S.l.: s.n.], 2005. p. 2828–2833.
- SILVA, A. R. D.; LAGES, W. S.; CHAIMOWICZ, L. Boids that see: Using self-occlusion for simulating large groups on gpus. **Comput. Entertain.**, ACM, New York, NY, USA, v. 7, n. 4, p. 51:1–51:20, jan. 2010. ISSN 1544-3574. Disponível em: <<http://doi.acm.org/10.1145/1658866.1658870>>.
- SINGH, S.; KAPADIA, M.; HEWLETT, B.; REINMAN, G.; FALOUTSOS, P. A modular framework for adaptive agent-based steering. In: **Symp. Int. 3D Graphics and Games**. [S.l.: s.n.], 2011. p. 141–150. ISBN 978-1-4503-0565-5.
- TECCHIA, F.; LOSCOS, C.; CONROY, R.; CHRYSANTHOU, Y. Agent behaviour simulator (abs): A platform for urban behaviour development. In: **In GTEC'2001**. [S.l.: s.n.], 2001.

THALMANN, D.; MUSSE, S. In: **Crowd Simulation**. [S.l.]: Springer London, 2013. ISBN 978-1-4471-4449-6.

TREUILLE, A.; COOPER, S.; POPOVIĆ, Z. Continuum crowds. In: **SIGGRAPH '06**. [S.l.]: ACM, 2006. p. 1160–1168. ISBN 1-59593-364-6.

TU, X.; TERZOPOULOS, D. Artificial fishes: physics, locomotion, perception, behavior. In: **Proceedings of the 21st annual conference on Computer graphics and interactive techniques**. New York, NY, USA: ACM, 1994. (SIGGRAPH '94), p. 43–50. ISBN 0-89791-667-0.

Ubisoft. **Assassin's Creed Syndicate**. 2015.

Warner Bros. PicturesMo. **The Hobbit: The Battle of the Five Armies**. 2014.

WARREN, C. Global path planning using artificial potential fields. In: **Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on**. [S.l.: s.n.], 1989. p. 316–321 vol.1.

WARREN, C. Multiple robot path coordination using artificial potential fields. In: **Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on**. [S.l.: s.n.], 1990. p. 500–505 vol.1.

WARREN, W.; FAJEN, B. Optic flow and beyond. In: VAINA, L. M.; BEARDSLEY, S. A.; RUSHTON, S. K. (Ed.). Norwell, MA, USA: Kluwer Academic Publishers, 2004. cap. From optic flow to laws of control, p. 307–337. ISBN 1-4020-2091-0.

XIONG, M.; LEES, M.; CAI, W.; ZHOU, S.; LOW, M. Y. H. “Hybrid modelling of crowd simulation”. **Procedia Computer Science**, v. 1, n. 1, p. 57 – 65, 2010. ISSN 1877-0509.

XU, S.; DUH, H. B.-L. A simulation of bonding effects and their impacts on pedestrian dynamics. **Intelligent Transportation Systems, IEEE Transactions on**, v. 11, n. 1, p. 153–161, 2010.

YERSIN, B.; MAÏM, J.; MORINI, F.; THALMANN, D. “Real-time crowd motion planning: Scalable avoidance and group behavior”. **Visual Computer**, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 24, n. 10, p. 859–870, 2008. ISSN 0178-2789.

ZHENG, X.; ZHONG, T.; LIU, M. “Modeling crowd evacuation of a building based on seven methodological approaches”. **Building and Environment**, v. 44, n. 3, p. 437 – 445, 2009. ISSN 0360-1323.

APÊNDICE A – PROJEÇÃO PANORÂMICA

A projeção panorâmica da técnica descrita neste trabalho é composta de várias projeções planares comuns. É preciso então encontrar os parâmetros para cada uma dessas projeções.

Uma transformação de projeção de *frustum* necessita de seis parâmetros para ser definida: esquerda (f_{left}), direita (f_{right}), topo (f_{top}), base (f_{bottom}), próximo (f_{near}) e distante (f_{far}). Os valores f_{near} e f_{far} são fornecidos como parâmetros de entrada, dependendo da configuração desejada.

Dado o número desejado de fatias s , e um ângulo de abertura horizontal α_h , tem-se que o ângulo horizontal de cada fatia α_h^{slice} será de:

$$\alpha_h^{slice} = \frac{\alpha_h}{s} \quad (\text{A.1})$$

Considerando que o *frustum* terá largura $2w$, pode-se calcular:

$$w = f_{near} \tan\left(\frac{\alpha_h^{slice}}{2}\right) \quad (\text{A.2})$$

Daí:

$$f_{left} = -w \quad (\text{A.3})$$

$$f_{right} = +w \quad (\text{A.4})$$

Tendo também como entrada o ângulo de abertura vertical α_v e o ângulo de orientação vertical r_v , pode-se como calcular a altura do *frustum*.

$$f_{top} = f_{near} \tan(\alpha_v + r_v) \quad (\text{A.5})$$

$$f_{bottom} = -f_{near} \tan(\alpha_v - r_v). \quad (\text{A.6})$$

Na técnica aplicada neste trabalho, todas as fatias são idênticas. Portanto, a mesma matriz de transformação de projeção pode ser usada para todas as sucessivas renderizações.

O que as torna diferentes entre si é a rotação em torno do eixo vertical do agente, que passa pela origem. Uma nova matriz de rotação é gerada para cada fatia, e é concatenada à projeção.

Como visto na Figura 39, o ângulo de rotação do *frustum* define a direção em que o seu eixo central estará apontando. Como desejamos que a face esquerda do primeiro *frustum* coincida com o plano esquerdo do campo de visão total, seu eixo central deve estar com ângulo β_0 :

$$\beta_0 = -\frac{\alpha_h}{2} + \frac{\alpha_h^{slice}}{2}. \quad (\text{A.7})$$

Como cada *frustum* consecutivo deve estar rotacionado α_h graus em relação ao anterior, temos:

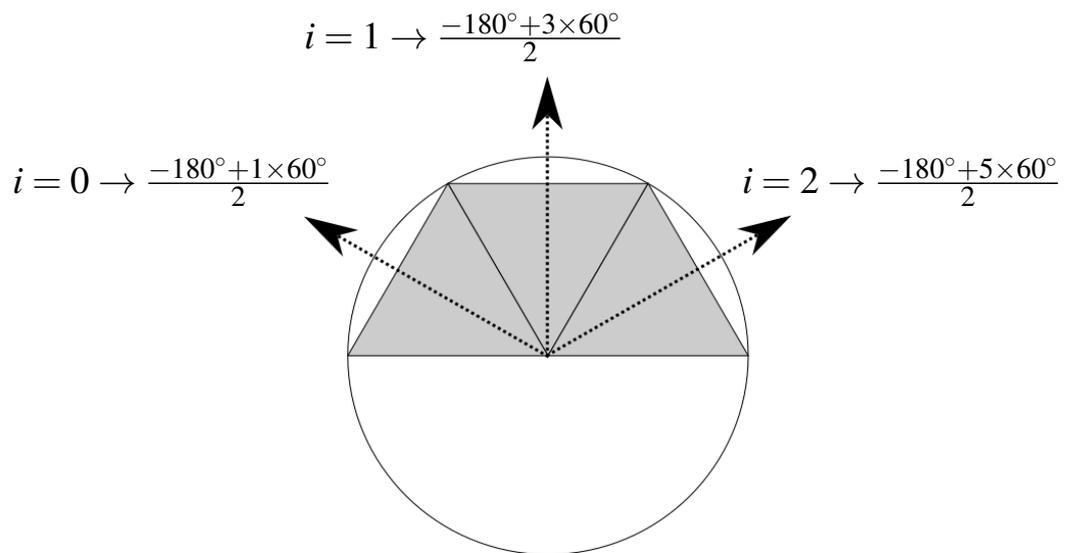
$$\beta_i = -\frac{\alpha_h^{slice}}{2} + \frac{\alpha_h^{slice}}{2} + (i\alpha_h) \quad (\text{A.8})$$

ou

$$\beta_i = -\frac{\alpha_h + (2i + 1)\alpha_h^{slice}}{2}. \quad (\text{A.9})$$

Se todos os agentes utilizarem as mesmas dimensões de visão, as matrizes só precisam ser calculadas uma única vez ao longo da execução do algoritmo.

Figura 39 – Ilustrando o resultado do algoritmo com 3 fatias e um ângulo total de 180° . Três frustums vão ser criados, com ângulo horizontal de 60° . O Valor indicado na imagem para cada frustum (0, 1 e 2) corresponde ao ângulo que a projeção é rotacionada em torno do eixo Y.



Fonte: Autor

APÊNDICE B – GRADIENTE DA FUNÇÃO DE CUSTO

A função de custo total é:

$$C_t = C_m + C_o + C_d. \quad (\text{B.1})$$

Deseja-se computar o gradiente nas dimensões de v_o , a velocidade do observador, e θ_o , a orientação do observador:

$$\nabla C_t = \frac{\partial C_t}{\partial v_o} dv_o + \frac{\partial C_t}{\partial \theta_o} d\theta_o. \quad (\text{B.2})$$

Para isso, precisamos computar as derivadas parciais:

$$\frac{\partial C_t}{\partial v_o} = \frac{\partial C_m}{\partial v_o} + \frac{\partial C_o}{\partial v_o} + \frac{\partial C_d}{\partial v_o} \quad (\text{B.3})$$

e

$$\frac{\partial C_t}{\partial \theta_o} = \frac{\partial C_m}{\partial \theta_o} + \frac{\partial C_o}{\partial \theta_o} + \frac{\partial C_d}{\partial \theta_o}. \quad (\text{B.4})$$

Em Dutra *et al.* (2015), o cálculo das derivadas parciais de C_m e C_o são detalhadas.

No Apêndice C será descrito como é calculada a derivada parcial da função de densidade.

APÊNDICE C – DERIVADAS PARCIAIS DA FUNÇÃO DE CUSTO

Tem-se função de custo de densidade, em termos do ângulo e da velocidade do agente observador, v_o e θ_o :

$$C_d(v, \theta_o) = k_1 e^{k_2 \|d_{o|i}^f\|^2} = k_1 e^{k_2 (d_{o|i}^f \cdot d_{o|i}^f)}. \quad (\text{C.1})$$

No modelo HSV, assim como no trabalho de Best *et al.* (2014), esses parâmetros eram dados por:

$$k_1 = \frac{1}{\sqrt{2\pi\sigma}} \quad (\text{C.2})$$

e

$$k_2 = \frac{-0.5}{\sigma^2}. \quad (\text{C.3})$$

No modelo GBM-D, esses parâmetros são separados para permitir melhor parametrização da função de custo, sendo k_1 e k_2 dois valores fornecidos diretamente pelo usuário.

A posição do agente observador o , depois de t segundos será:

$$p_o^t = p_o + v_o t, \quad (\text{C.4})$$

e analogamente, a posição do obstáculo i será:

$$p_i^t = p_i + v_i t, \quad (\text{C.5})$$

onde

$$v_o = (v \cos(\theta_o), v \sin(\theta_o)) \quad (\text{C.6})$$

e

$$\hat{v}_o = (\cos(\theta_o), \sin(\theta_o)). \quad (\text{C.7})$$

p_i , p_o e v_i são constantes para cada execução do algoritmo.

A distância entre os dois será de:

$$d_{o|i}^t = p_i^t - p_o^t. \quad (\text{C.8})$$

A distância dos agentes na direção da velocidade é dada por:

$$d_{o|i}^v = (d_{o|i}^t \cdot \hat{v}_o) \hat{v}_o, \quad (\text{C.9})$$

sendo possível, assim, deslocar os agentes na direção perpendicular à velocidade:

$$d_{o|i}^f = 2.5d_{o|i}^t - 1.5d_{o|i}^v. \quad (\text{C.10})$$

Derivando a função de custo em uma variável γ :

$$\frac{\partial C}{\partial \gamma} = 2k_1 k_2 e^{k_2(d_{o|i}^f \cdot d_{o|i}^f)} (d_{o|i}^f \cdot \frac{\partial d_{o|i}^f}{\partial \gamma}) = 2k_2 C(d_{o|i}^f \cdot \frac{\partial d_{o|i}^f}{\partial \gamma}), \quad (\text{C.11})$$

$$\frac{\partial d_{o|i}^f}{\partial \gamma} = 2.5 \frac{\partial d_{o|i}^t}{\partial \gamma} - 1.5 \frac{\partial d_{o|i}^v}{\partial \gamma}, \quad (\text{C.12})$$

$$\frac{\partial d_{o|i}^t}{\partial \gamma} = \frac{\partial p_i^t}{\partial \gamma} - \frac{\partial p_o^t}{\partial \gamma} = -\frac{\partial v_o}{\partial \gamma} t, \quad (\text{C.13})$$

$$\frac{\partial d_{o|i}^v}{\partial \gamma} = \hat{v}_o (\hat{v}_o \cdot \frac{\partial d_{o|i}^t}{\partial \gamma} + d_{o|i}^t \cdot \frac{\partial \hat{v}_o}{\partial \gamma}) + (d_{o|i}^t \cdot \hat{v}_o) \frac{\partial \hat{v}_o}{\partial \gamma}. \quad (\text{C.14})$$

Derivando v_o em v temos:

$$\frac{\partial v_o}{\partial v} = (\cos(\theta_o), \sin(\theta_o)), \quad (\text{C.15})$$

$$\frac{\partial \hat{v}_o}{\partial v} = (0, 0). \quad (\text{C.16})$$

Derivando v_o em θ_o temos:

$$\frac{\partial v_o}{\partial \theta_o} = (-v \sin(\theta_o), v \cos(\theta_o)), \quad (\text{C.17})$$

$$\frac{\partial \hat{v}_o}{\partial \theta_o} = (-\sin(\theta_o), \cos(\theta_o)). \quad (\text{C.18})$$