

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**CONTROLE DE COMPORTAMENTOS
DE GRUPOS DE PERSONAGENS
VIRTUAIS**

RAFAEL ARAÚJO RODRIGUES

Dissertação de Mestrado apresentada como requisito para obtenção do título de Mestre em Ciência da Computação pelo Programa de Pós-graduação da Faculdade de Informática. Área de concentração: Ciência da Computação.

Orientadora: Soraia Raupp Musse

Porto Alegre, Brasil
2010

Dados Internacionais de Catalogação na Publicação (CIP)

R696c Rodrigues, Rafael Araújo
Controle de comportamentos de grupos de personagens
virtuais / Rafael Araújo Rodrigues. – Porto Alegre, 2010.
85 f.

Diss. (Mestrado) – Fac. de Informática, PUCRS.
Orientador: Prof^ª. Dr^ª. Soraia Raupp Musse.

1. Informática. 2. Computação Gráfica. 3. Simulação e
Modelagem em Computadores. I. Musse, Soraia Raupp.
II. Título.

CDD 006.6

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**Controle de Comportamentos de Grupos de Personagens Virtuais**", apresentada por Rafael Araújo Rodrigues, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas Interativos de Visualização, aprovada em 27/03/09 pela Comissão Examinadora:

Prof. Dra. Soraia Raupp Musse -
Orientador(a)

PPGCC/PUCRS

Prof. Dr. Osmar Norberto de Souza -

PPGCC/PUCRS

Prof. Dr. Bruno Feijó -

PUCRJ

Homologada em 04/04/2012, conforme Ata No. 01/2012 pela Comissão Coordenadora.

Prof. Dr. Fernando Gehm Moraes
Coordenador.

PUCRS

Campus Central

Av. Ipiranga, 6681 - P32 - sala 507 - CEP: 90619-900

Fone: (51) 3320-3611 - Fax (51) 3320-3621

E-mail: ppgcc@pucrs.br

www.pucrs.br/facin/pos

*“A journey of a thousand miles
begins with a single step.”
Lao-tzu, The Way of Lao-tzu*

Agradecimentos

Aos amigos que me acompanharam durante este tempo e a minha namorada Tati.

Controle de comportamentos de grupos de personagens virtuais

RESUMO

Este trabalho descreve um modelo para gerar comportamentos guiados de grupos de personagens inspirado biologicamente no algoritmo de colonização do espaço para simulação de multidões *Biocrowds*.

A proposta consiste em criar uma extensão para o *BioCrowds* com o objetivo de simular comportamentos guiados conforme descrito por Craig Reynolds. Esta abordagem é chamada *Tree Paths* e é responsável pelo controle de movimentos de personagens de forma robusta e realística.

Enfim, são estudados e implementados alguns comportamentos que também complementam o *BioCrowds*, os quais, denominam-se de alinhamento e formação, que constituem uma classe de comportamentos emergentes útil em computação gráfica e entretenimento.

Palavras-chave: *steering behaviors*, computação gráfica, simulação.

CONTROLLING BEHAVIORS FROM GROUPS OF VIRTUAL CHARACTERS

ABSTRACT

This work describes a model for generating *steering behaviors* groups of virtual characters based on the biologically-motivated space colonization algorithm in the crowd simulation model *BioCrowds*.

The proposition is to create an extension for *BioCrowds* in order to simulate *steering behaviors* according to the concept proposed by Craig Reynolds. This approach is named *Tree Paths* and is responsible for the motion control of characters providing robust and realistic group behaviors.

Additionally, some extra behaviors were also studied and implemented in *BioCrowds*. These are called *alignment* and *formation*, and consist in a small class of emergent behaviors that can also be useful in computers graphics and entertainment.

Keywords: *steering behaviors*, computer graphics, simulation.

Lista de Figuras

| | | |
|------|--|----|
| 2.1 | Imagem do filme Stanley and Stella, gerado usando o modelo Reynolds [REY87], onde pássaros sobrevoam um cardume. Fig. extraída de [REY87]. | 28 |
| 2.2 | Alargamento do corredor causado por dispersão. Extraído de [HEL00] | 29 |
| 2.3 | Multidão reagindo em <i>ViCrowd</i> . Extraído de [MUS01] | 30 |
| 2.4 | Diferentes maneiras de caminhar resultantes de diferentes emoções. Extraído de [MUS01]. | 30 |
| 2.5 | Área de atração e repulsão em <i>ViCrowd</i> . Extraído de [MUS01]. | 31 |
| 2.6 | Diferentes visões da simulação de uma cidade em <i>Continuum Crowds</i> . Extraído de [TEU06]. | 32 |
| 2.7 | Camadas de um sistema de comportamento de um agente autônomo. Extraído de [REY99]. | 33 |
| 2.8 | Demonstração dos comportamentos <i>Seek/Flee</i> proposto por Reynolds. Extraído de [REY99]. | 34 |
| 2.9 | Demonstração dos comportamentos <i>pursuit/evasion</i> propostos por Reynolds. Extraído de [REY99]. | 35 |
| 2.10 | Foto sintética que ilustra, no destaque, o algoritmo inspirado biologicamente para canalização de nervuras. Extraído de [RUN05]. | 36 |
| 2.11 | Modelo de funcionamento do algoritmo de canalização das nervuras. Extraído de [RUN05]. | 37 |
| 2.12 | Exemplo de sequências de geração de nervuras nas folhas, simulação com diferentes configurações de parâmetros. Extraído de [RUN05]. | 37 |
| 3.1 | Exemplo de agente contornando uma região sem marcadores | 41 |
| 3.2 | Área pessoal (círculo) e marcadores (pontos) associados a cinco agentes (quadrados). Os marcadores capturados por cada agente são mostrados com a mesma cor. | 41 |
| 3.3 | Agentes com linhas relacionando estes com seus respectivos marcadores em uma visualização 2D. | 42 |
| 4.1 | Comparação entre o algoritmo de canalização de nervuras e o algoritmo <i>Tree Paths</i> . Inicializações, crescimento e os seus devidos resultados. | 47 |

| | | |
|-----|---|----|
| 4.2 | O agente tenta chegar ao seu objetivo ilustrado pela bandeira vermelha. Os pontos mais escuros são nodos chaves onde o agente toma uma decisão de qual caminho seguir. Já os pontos mais claros, representam nodos criados ao longo das iterações onde a ligação entre eles é um segmento de nodo. | 49 |
| 4.3 | Novos nodos são gerados e restringem os marcadores dentro da "região de restrição" | 51 |
| 4.4 | Exemplo de aplicação de <i>Tree Paths</i> na perseguição de agente. Os marcadores são pontos na superfície. As regiões em preto são regiões sem marcadores. A Figura mostra a árvore desenvolvida de um agente no intuito de chegar até o outro agente. . | 52 |
| 4.5 | Superfície discretizada por meio de marcadores em <i>Tree Paths</i> | 53 |
| 4.6 | O problema da perseguição de alvos móveis | 54 |
| 4.7 | Solução para o problema do alvo móvel | 54 |
| 4.8 | Demonstração do que acontece com a simulação quando um agente está habilitado a eliminar os nodos de outro agente, dentro de um raio. O agente mais abaixo no primeiro quadro, conforme se desloca para cima, elimina alguns nodos do agente mais a esquerda que tenta chegar até a bandeira. Nota-se que no decorrer do processo, a árvore do agente volta a crescer por outros ramos. | 56 |
| 4.9 | Agente (central) fugindo de 3 perseguidores colaborativos. | 57 |
| 5.1 | Exemplos de formações de agentes em cima de regiões | 60 |
| 5.2 | Agentes chegando em uma área de formação que ilustram "VH". | 61 |
| 5.3 | Alinhamento entre agentes. | 62 |
| 5.4 | Exemplo de máscara de pesos, que fará os agentes colaborarem para alinhar-se na horizontal e vertical. | 63 |
| 5.5 | Resultado da simulação após agentes se alinharem. | 63 |
| 6.1 | Problema em <i>Tree Paths</i> com marcadores oclusos por obstáculos. | 66 |
| 6.2 | Comparação proporcional da média de marcadores por metro quadrado em <i>BioCrowds</i> (à esquerda sendo $17.5m^2$) e <i>Tree Paths</i> (à direita sendo $0.4m^2$) usadas nas simulações. | 67 |
| 6.3 | Comparação da distância percorrida entre A-Estrela (acima) e <i>Tree Paths</i> (abaixo). Na primeira simulação foi percorrida uma distância de $78.05 m$, e na segunda, o resultado foi $83.60 m$ percorridos. | 69 |
| 6.4 | Perseguição de alvo móvel. O agente à esquerda persegue o agente à direita que vaga pelo ambiente. | 72 |
| 6.5 | Planejamento de rota em <i>Tree Paths</i> . O agente é o cubo em preto, os obstáculos são representados como retângulos. | 73 |
| 6.6 | Perseguição colaborativa de alvo móvel. Os agentes à esquerda colaboram para alcançar o agente à direita. | 74 |

| | | |
|------|---|----|
| 6.7 | Comportamento de Fuga, agente em azul tenta escapar da área que circunda outros agentes que permanecem estáticos. | 75 |
| 6.8 | Colaboração entre agentes que perseguem um agente em fuga. Agentes à esquerda colaboram tentando alcançar o agente a direita que tenta fugir. | 76 |
| 6.9 | Sequência de imagens que mostram agentes chegando a uma área de formação. | 77 |
| 6.10 | Sequência de imagens que mostram o alinhamento entre agentes. | 78 |

Sumário

| | |
|---|-----------|
| Lista de Figuras | 15 |
| 1 Introdução | 21 |
| 1.1 O problema | 22 |
| 1.2 Objetivo | 22 |
| 1.2.1 Objetivos específicos | 23 |
| 1.3 Organização da dissertação | 23 |
| 2 Trabalhos relacionados | 25 |
| 2.1 Simulação de multidões | 25 |
| 2.1.1 Modelos mais consolidados de Simulação de Multidões | 26 |
| 2.2 <i>Steering Behaviors</i> | 32 |
| 2.3 Comportamentos básicos classificados como <i>Steering Behaviors</i> | 33 |
| 2.3.1 Chegar e evitar (<i>Seek / Flee</i>) | 33 |
| 2.3.2 Perseguir e fugir (<i>pursuit/evasion</i>) | 34 |
| 2.4 O modelo de canalização de nervuras de plantas | 35 |
| 3 Fundamentação do modelo BioCrowds | 39 |
| 3.1 Adaptação do modelo biológico para simulação de multidões | 39 |
| 3.1.1 Descrição do modelo | 40 |
| 4 Steering behaviors e BioCrowds | 45 |
| 4.1 O modelo <i>Tree Paths</i> | 46 |
| 4.1.1 Adaptação do modelo de canalização de nervuras para o <i>Tree Paths</i> | 46 |
| 4.1.2 Definições para o modelo <i>Tree Paths</i> | 48 |
| 4.2 Integrando <i>Tree Paths</i> e <i>BioCrowds</i> | 51 |
| 4.3 Perseguição de alvos móveis com <i>Tree Paths</i> | 52 |

| | | |
|----------|---|-----------|
| 4.4 | Colaboração entre agentes | 53 |
| 4.5 | Comportamento de fuga | 55 |
| 4.6 | União dos comportamentos colaboração e fuga | 55 |
| 5 | Comportamentos emergentes de grupos em BioCrowds | 59 |
| 5.1 | Formação | 59 |
| 5.2 | Alinhamento entre agentes | 60 |
| 6 | Resultados | 65 |
| 6.1 | Considerações sobre parâmetros usados | 65 |
| 6.2 | Simulações realizadas | 68 |
| 7 | Considerações Finais | 79 |
| | Bibliografia | 83 |

Capítulo 1

Introdução

A área de simulação de multidões visa o realismo dos resultados tanto numéricos quanto visuais. Os desafios são grandes, e entre eles estão, movimentação individual dos agentes, tratamento de colisões, comportamento de grupos, simulação afetiva, situações de emergência, etc. Apesar de vários modelos já terem sido propostos, nenhum foi capaz de obter bons resultados em situações genéricas. Para aplicações, em computação gráfica e realidade virtual, é necessário que a modelagem de agentes com níveis de autonomia seja suficiente para simular um "teatro improvisado" ou o comportamento de quaisquer entidades simuladas, sejam humanos, animais ou seres inanimados.

Por outro lado, a movimentação de agentes em ambientes virtuais pode ser considerada parte primordial do sistema de inteligência artificial de um game, ou ainda, importante na dinâmica e no desafio imposto ao jogador. Para Reynolds [REY99], estes agentes podem ter um caráter deliberativo ou reativo em suas ações. Além disso, o comportamento de agentes autônomos pode servir como aplicação na área de segurança, uma vez que modelos de simulação de multidões são usados para realização de simulação de evacuações ou emergência em projetos arquitetônicos, também podem ser incorporados em softwares interativos que servem como simuladores para treinamento, com aplicação na área de defesa.

Apesar das pesquisas na área, ainda é um desafio tentar simular corretamente o tratamento de colisão entre os personagens. Alguns modelos conhecidos na literatura propuseram diferentes soluções para o problema, conforme destacado a seguir: Reynolds [REY87] descreveu o primeiro modelo comportamental distribuído onde simulava pássaros, rebanhos e cardumes, por meio de um modelo de regras e percepção dos agentes. Helbing et al. [HEL00], propuseram um modelo de forças físicas e sócio-psicológicas para descrever o comportamento de multidões humanas em situações de pânico. Ele considerou cada indivíduo como sendo uma partícula, sujeita às forças de repulsão, atração e restauradoras de objetivo, presentes no seu modelo. Treuille e Popovic [TEU06] usaram dinâmica de fluidos para construção de campos potenciais, simulando os comportamentos das multidões a partir de diferentes densidades em tempo real.

Embora os modelos citados, assim como outros existentes na literatura, obtenham bons resultados, a colisão entre humanóides, em altas densidades, ainda é um problema em aberto. Assim, nesta

dissertação procurou-se inspiração em outras áreas do conhecimento, de maneira a reproduzir um comportamento realista e não determinístico. Uma das contribuições deste trabalho é prover um modelo para simulação de multidões baseado em biologia. Durante o período deste trabalho, foi desenvolvido um modelo denominado *BioCrowds*, baseado no algoritmo de canalização de nervura de plantas [RUN05], capaz de simular uma série de comportamentos emergentes, observados nas multidões.

Como extensão deste modelo, propôs-se um conjunto de comportamentos de controle de grupos (*steering behaviors*), que são apresentados na literatura [REY99], como básicos para a locomoção de agentes em ambientes virtuais. Estes comportamentos são de extrema importância para o realismo das simulações. A abordagem proposta é denominada *Tree Paths* [ROD09], um algoritmo de planejamento de rota baseado em colonização do espaço.

Tree Paths integrado com o *BioCrowds*, propicia ao agente analisar o ambiente com uma percepção mais abrangente. Desta maneira, os modelos assumem tarefas mais globais do agente, por exemplo, definição de rota, enquanto *BioCrowds* assume um caráter mais local, como variação de velocidade e evitar colisão. O controle global é essencial para que se possa reproduzir os comportamentos guiados conhecidos como *steering behaviors*, pois é por meio deles que o agente se torna capaz, por exemplo, de fugir de algum lugar, perseguir um alvo, achar um caminho ou uma porta em um ambiente complexo. Por fim, foram abordados também um conjunto de comportamentos emergentes denominados *formação e alinhamento* entre agentes, propondo modificações em *BioCrowds*.

1.1 O problema

Desenvolver um modelo de simulação de personagens, robusto e eficiente para diversas situações, ainda é uma tarefa árdua para os pesquisadores da área. Diferentes técnicas já foram usadas pelos pesquisadores, inspiradas em sistemas de partículas, fluidos, entre outras, e serão apresentados no Capítulo 2. No entanto, estes modelos não são genéricos o suficiente para simular uma vasta gama de aplicações. Além disso, os modelos tendem a crescer em sua complexidade, e o número grande de parâmetros implica em diferentes calibrações dos mesmos para realização de simulações às vezes parecidas

1.2 Objetivo

Em um primeiro momento, desenvolveu-se no âmbito do laboratório VHLab, o modelo *BioCrowds*, um modelo para simulação de multidões capaz de simular uma grande gama de comportamentos inerentes e emergentes de uma multidão. Este foi inspirado na teoria da competição por espaço ([RUN05]), observada na dinâmica da canalização das nervuras das plantas, tornando o *BioCrowds*, um modelo de simulação de multidões com inspiração biológica. Explorando o modelo *Bi-*

oCrowds, buscou-se incluir novas propriedades neste a fim de torná-lo capaz de simular um conjunto de comportamentos guiados conforme definidos por Reynolds em [REY99] como *steering behaviors*.

Desta forma, o principal objetivo deste trabalho é a reprodução dos *steering behaviors* em *BioCrowds*. A abordagem feita foi nomeada *Tree Paths* [ROD09], e propõem um modelo de navegação em ambientes virtuais, utilizando um algoritmo de colonização do espaço.

1.2.1 Objetivos específicos

Entre os objetivos específicos desta dissertação, estão:

- realizar estudo do estado da arte na área de simulação de multidões;
- elaborar um modelo de simulação de multidões baseado no algoritmo de colonização de espaço [RUN05], (em conjunto com outros pesquisadores do VHLab);
- avaliar os resultados do modelo;
- expandir o modelo incluindo uma camada responsável pelos comportamentos guiados (*steering behaviors*);
- desenvolver um protótipo;
- testar e avaliar os resultados.

1.3 Organização da dissertação

Este trabalho foi desenvolvido dentro do laboratório VHLab, no Projeto VHSimul, no PPGCC na PUC, e contou com o financiamento da HP Brazil.

Inicialmente, este trabalho tem como área principal a computação gráfica, encaixando-se em uma subárea conhecida como animação comportamental. Após uma introdução ao tema no Capítulo 1, é apresentado, no Capítulo 2, uma revisão bibliográfica sobre o tema incluindo os principais trabalhos para simulação de multidões, bem como alguns fundamentos para o desenvolvimento deste trabalho a saber: a conceituação dada por Reynolds para comportamentos guiados de grupos (*steering behaviors*) e o algoritmo de canalização de nervuras que serviu de inspiração para este trabalho. No Capítulo 3 é apresentado o modelo para simulação de multidões de inspiração biológica *BioCrowds* desenvolvido no âmbito do laboratório VHLab. Em seguida, no Capítulo 4, é apresentado o modelo *Tree Paths* e como este foi integrado ao *BioCrowds* para prover controle global de grupos de personagens. Enfim, no Capítulo 6 são apresentados os resultados obtidos, e no Capítulo 7 algumas considerações finais e problemas ainda em aberto.

Capítulo 2

Trabalhos relacionados

Nesta seção são abordados os trabalhos e modelos que serviram como base para esta pesquisa e entendimento do estado da arte. São abordados os principais modelos para simulação de multidões, categoria a qual *BioCrowds* pertence, o conceito de *steering behaviors*, abordado por Reynolds, ao qual esta dissertação se propõe a estender *BioCrowds* para simulação destes. E o algoritmo de canalização de nervuras [RUN05] que serviu como inspiração, tanto para *BioCrowds*, quanto para a extensão *Tree Paths* proposta nesta dissertação.

2.1 Simulação de multidões

Nesta seção alguns trabalhos serão brevemente descritos. No entanto, alguns deles, pela importância no desenvolvimento deste, serão detalhados posteriormente. Alguns importantes trabalhos relacionados à área de simulação comportamental de grupos, contexto no qual o presente trabalho está inserido, são a seguir apresentados. Reynolds [REY87] descreveu o primeiro modelo comportamental distribuído para simular bandos de pássaros, rebanhos e cardumes formados por agentes com habilidades de percepção. Nesse modelo, os animais virtuais mantêm posição e orientação próprias dentro do grupo, contrabalançando seu desejo de evitar colisão com os demais, igualar sua velocidade à de seus vizinhos e dirigir-se para o centro do grupo.

Helbing e seus colaboradores [HEL03], propuseram um modelo de forças físicas e sócio-psicológicas a fim de descrever o comportamento de multidões humanas em situações de pânico. Em 1994, Still [STI94] desenvolveu um conjunto de programas denominado LEGION para analisar a dinâmica de uma multidão em situações de evacuação de ambientes restritos e complexos, utilizando um algoritmo de "mínimo esforço". Nesse sistema são consideradas quatro regras comportamentais: objetivo, mobilidade, limitação e assimilação. Para evitar o comportamento homogêneo da população, foi introduzido um ruído na velocidade dos agentes. O sistema foi utilizado para simular multidões em vários locais e eventos como as Olimpíadas de Sydney, por exemplo.

Bouvier e seus colaboradores [BOV96], usaram um sistema de partículas adaptadas para o estudo

de multidões humanas, no qual as pessoas são modeladas como um grupo de partículas interagentes. O seu movimento, assim como seus objetivos e decisões, são baseados em forças newtonianas. Goldenstein et al. [GOL01] desenvolveram um modelo dinâmico não linear para simular grandes quantidades de agentes em um ambiente que pode ser alterado em tempo real. Esse modelo contempla comportamentos de baixo nível dos agentes, tal como evitar colisão e encontrar um caminho até um alvo fixo ou móvel.

Thomas e Donikian [DON00] apresentam um modelo de ambiente urbano virtual usando estruturas e informações apropriadas para animação comportamental. Foi projetado para este trabalho um modelador de cidades que permite a produção automática de ambientes urbanos complexos para realizarem-se animações comportamentais. O principal objetivo é simular vida em ruas e calçadas e não dentro de prédios e locais fechados. O principal interesse é buscar informações das construções baseado no seu exterior, definindo suas funcionalidades (se são casas, lojas, etc) e os dados de entrada e saída do fluxo dos pedestres dependendo da hora do dia.

Musse e Thalmann [MUS97] propuseram um modelo hierárquico para descrever multidões com diferentes níveis de autonomia: guiados, programados ou autônomos. Nesse modelo, foram utilizados conceitos sociológicos para representar comportamentos específicos. Musse apresentou o *ViCrowd*, um sistema que gera automaticamente multidões humanas baseadas em grupos. O comportamento da multidão é obtido por meio de regras estabelecidas de acordo com as informações contidas nos grupos de indivíduos.

Ming Lin [LIN07] apresentou um novo algoritmo para navegação de um grande número de agentes independente da complexidade e dinamismo do ambiente. Propôs a representação chamada "*Adaptive Elastic Roadmaps*" (*AERO*) que realiza a conectividade e deformação de um grafo baseado na interação entre agentes e movimentação dos obstáculos.

Lavalle [LAV00] apresentou um algoritmo de conceito semelhante ao usado nesta dissertação intitulado *Rapidly-exploring Random Tree (RRT)*, que com uma estrutura de dados aleatória designada para uma classe de problemas de rotas, obteve bons resultados para definição de geração de rotas, através da geração de árvores aleatórias representadas por grafos. No entanto, de maneira geral, o algoritmo é insuficiente para resolver o problema de encontrar o melhor caminho, mas pode ser um componente importante para a definição deste.

2.1.1 Modelos mais consolidados de Simulação de Multidões

Como toda a pesquisa de cunho científico, a criação de modelos de simulação é fortemente embasada em trabalhos já realizados. Alguns trabalhos pioneiros da área mostraram que esse processo não é fácil, e sua complexidade pode aumentar conforme são inseridas características no modelo. Os trabalhos descritos nessa seção representam a evolução nesse processo, apresentada na literatura. Para tanto, foram selecionados quatro modelos considerados básicos para a introdução da área de modelos de simulação, são eles: *flock boids* de Reynolds [REY87] que implementou um sistema

comportamental a partir de dinâmica de partículas, Helbing [HEL00] que transformou forças físicas em forças sociológicas, Musse [MUS01], que tratou sobre o problema de níveis de autonomia e complexidade como fator crítico para performance, e Treuille [TEU06] que usou dinâmica de fluidos para construção de campos potenciais para simulação de altas densidades em tempo real.

Flocks, Herds, and Schools de Reynolds

Em 1987, Craig Reynolds [REY87] publica um artigo pioneiro sobre a dinâmica de pássaros e peixes (ver Figura 2.1) simulando um comportamento de grupos em um modelo de movimento polarizado, agregado e evitando colisão. O modelo simula pássaros e peixes, onde cada pássaro ou peixe é um agente autônomo que se comporta de acordo com sua própria percepção do ambiente dinâmico. Este modelo apresentou resultados bastante relevantes para a área, sendo certamente um dos modelos que mais gerou impacto em termos de algoritmos implementados e aplicados em áreas, como jogos e cinema. O comportamento emerge em função das interações de três regras principais:

- evitar colisão;
- igualar a velocidade com os vizinhos; e
- tentar ir ao centro do grupo.

O modelo de Reynolds propõe a implementação que inclui regras locais de comportamento, conforme especificado acima. É usada uma abordagem de partículas inspirada no trabalho de Reeves [REE83], onde cada uma é representada por um objeto modelado geometricamente, tornando mais complexa a estrutura de cada partícula, tanto em termos de movimentação, quanto no tratamento das colisões.

Os resultados de cada uma das regras locais de comportamento são recombinados, em tempo de processamento e em função dos dados percebidos por cada entidade da simulação. Este processo acontece em todos os vizinhos de cada entidade, permitindo resultados mais coerentes e realistas.

Prevenção de colisão e a dinâmica de nivelamento de velocidade são aspectos complementares. Juntos, eles asseguram que os agentes da simulação sejam livres para se movimentar, e ao mesmo tempo, mantenham a integridade do bando sem a necessidade de que um vá ao encontro do outro. A prevenção de colisão é um dos mais difíceis comportamentos ao tratar-se de um eminente impacto. A colisão estática, descrita em seu modelo, é baseada na posição relativa do bando, desconsiderando suas velocidades. Da mesma forma, o nivelamento da velocidade é feito desconsiderando a posição.

Se um pássaro, ou indivíduo, não consegue êxito no nivelamento de sua velocidade com seus vizinhos, este colidirá com seu vizinho em um curto espaço de tempo. Com o processo de nivelamento da velocidade, as distâncias entre os pássaros são também niveladas. A prevenção de colisão estática serve para estabelecer a distância mínima requerida, sendo assim, a equivalência das velocidades tende a manter este comportamento, evitando colisões.

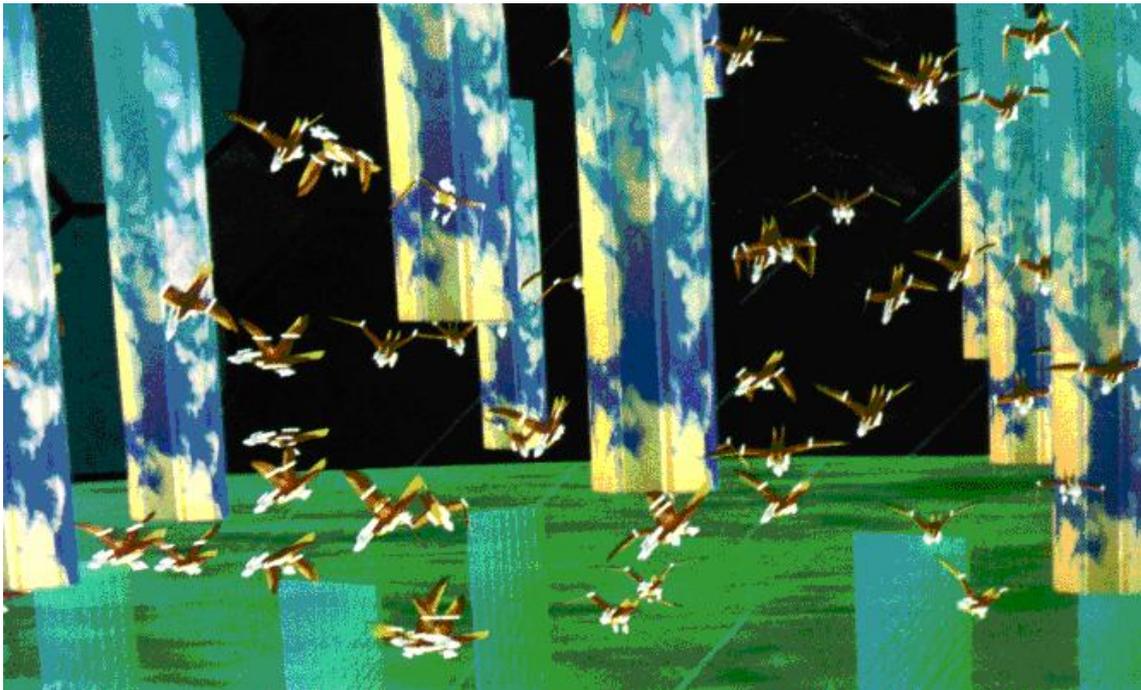


Figura 2.1: Imagem do filme Stanley and Stella, gerado usando o modelo Reynolds [REY87], onde pássaros sobrevoam um cardume. Fig. extraída de [REY87].

Às vezes, bandos reais dividem-se em grupos com o fim de desviar-se de obstáculos. Para ser realista, o modelo também precisa simular este efeito. O conceito de *flock* não despreza este processo, uma vez que as entidades que fazem parte do mesmo *flock*, respeitam as regras locais do *flock*, enquanto não houver colisões com obstáculos.

Dinâmicas de pedestres de Helbing

Helbing [HEL00] propôs um modelo (Figura 2.2), também baseado em sistema de partículas, para representar o fenômeno de pânico em situações de emergência. Suas simulações da dinâmica de multidões são baseadas em modelos de força generalizada, que se aplicam a descrição das formações e movimentações das pessoas, observadas durante casos de pânics.

Em seus modelos, Helbing destaca sua abordagem por meio de forças sociais traduzidas em equações da dinâmica, que caracterizam comportamentos automáticos das partículas. Essas equações constituem em definições como "forças sociais" que descrevem a influência do ambiente nos pedestres. A equação de movimentação (distinta para cada indivíduo), além de métodos de flutuação, correspondem à equações que definem o comportamento dos pedestres com um certo nível de aleatoriedade. As forças sociais nos modelos de Helbing operam baseadas em três princípios.

- Pedestres tentam caminhar em uma direção desejada em certa velocidade;
- Pedestres mantêm-se distantes das bordas (construções, paredes, ruas, obstáculos), em um

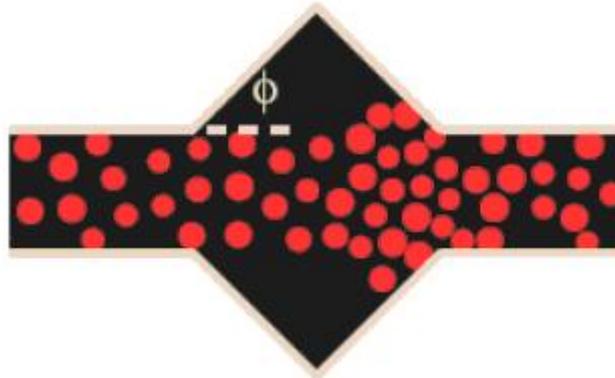


Figura 2.2: Alargamento do corredor causado por dispersão. Extraído de [HEL00]

efeito de repulsividade;

- A movimentação de um pedestre é influenciada pelos pedestres próximos.

Estas forças permitem ao indivíduo a realização de "manobras evasivas" ou diminuição de velocidades, a fim de manter uma distância coerente dos outros indivíduos.

Modelo hierárquico de Musse - ViCrowd

O modelo proposto por Musse et al [MUS01], consolidou a ferramenta *ViCrowd* (Figura 2.3) baseada na conjunção de abordagens de modelos mais tradicionais, propondo o conceito de níveis de autonomia em multidões. Este modelo hierárquico inova em como relacionar a multidão em uma estrutura hierárquica que vai de um indivíduo, aos grupos e a multidão inteira. Neste ponto, defende-se a idéia de que um sistema deve ter níveis de complexidade para criação de animações realísticas, onde, um indivíduo é o elemento mais simples e a multidão é o mais complexo.

Dependendo da complexidade proposta pela simulação, simples comportamentos podem ser suficientes. No entanto, regras de comportamento mais complexas podem ser necessárias, neste caso, é possível incluí-las na simulação, aprimorando seu realismo.

O presente modelo possibilita o controle de multidões de diferentes maneiras, podendo ser guiado, programado ou por meio de regras de comportamento, baseadas em eventos e reações. Na Figura 2.4 pode ser visto diferentes maneiras de caminhar resultante de distintas emoções em *ViCrowd*.

ViCrowd apresenta um modelo para geração automática de multidões baseadas em grupos, ao invés de agentes individuais. Comparando com outros trabalhos no domínio de "autonomia" e "inteligência", pode-se dizer que estes são sempre baseados em indivíduos, enquanto *ViCrowd* é baseado no grupo.

No caso de *ViCrowd* os grupos são estruturas mais "inteligentes", enquanto indivíduos seguem as especificações do grupo. Esta decisão é devido a necessidade de execução em tempo real, onde



Figura 2.3: Multidão reagindo em *ViCrowd*. Extraído de [MUS01]



Figura 2.4: Diferentes maneiras de caminhar resultantes de diferentes emoções. Extraído de [MUS01].

estas características tem a função de otimizar as informações necessárias para os agentes inteligentes. *ViCrowd* ainda é baseado em *Flocking Systems* [REY87], mas também inclui simples definições de regras de comportamento usando eventos condicionais e reações. Na Figura 2.5 pode ser visto a ação de uma área de atração e repulsão em *ViCrowd*.

Define-se como nível de autonomia a capacidade do agente criar comportamentos de forma autônoma. Em *ViCrowd*, a abordagem de controle possui três níveis, guiados, programados e autônomos, indo da total interatividade em agentes guiados, a total autonomia em agentes autônomos. Multidões guiadas são dirigidas usando um controle externo para interação (mouse, teclado, etc). Ainda, *ViCrowd* pode ser parte de um sistema cliente/servidor, que é chamado de *City-Server*, dedicado ao tratamento das multidões. Alguns clientes diferentes deste *framework* podem ser implementados independentemente de *ViCrowd*. Multidões programadas, fazem parte da implementação de regras baseadas em scripts, estes scripts definem eventos e como reagir a cada estímulo específico, sem necessidade de intervenção do usuário durante a simulação. Já em agentes autônomos, comportamentos emergentes surgem de regras definidas, baseadas em uma base de conhecimento e informação do ambiente.

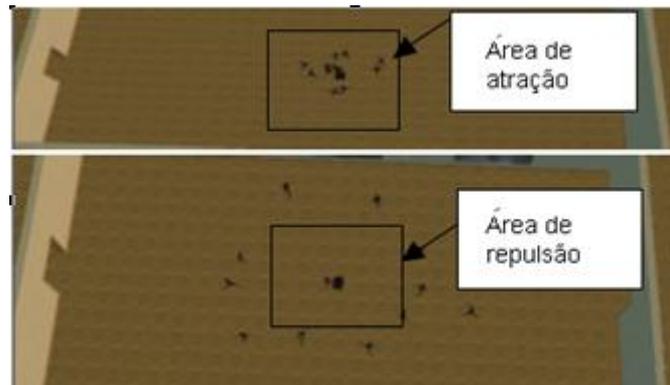


Figura 2.5: Área de atração e repulsão em *ViCrowd*. Extraído de [MUS01].

***Continnum Crowds* de Treuille**

Continnum Crowds (Figura 2.6) é um dos últimos trabalhos importantes da área na literatura. Zoran Popovic apresenta uma abordagem para grandes quantidades de indivíduos usando um método baseado em sistema de partículas [TEU06]. Por meio de conceitos de dinâmicas de fluidos, o autor propõe a construção de campos potenciais que são atualizados conforme a direção, fluxo e densidade de indivíduos naquela região. A dinâmica do modelo construído funciona por intermédio de três regras fundamentais:

- Todo indivíduo tenta chegar a um objetivo.
- Os indivíduos tentam mover-se na maior velocidade possível, havendo um limite de velocidade imposta no ambiente; Esse princípio levou os autores a um equacionamento que permita com que os indivíduos optem por caminhos em que possam manter velocidade máxima, evitando caminhos congestionados;
- Os indivíduos tentam evitar campos de desconforto, que podem ser uma região com alta densidade de personagens, obstáculos, etc.

Estas idéias sumarizam a escolha do caminho por um indivíduo, tentando minimizar a combinação linear entre três termos:

- Tamanho do caminho;
- Quantidade de tempo até destino;
- Desconforto percebido, por unidade de tempo, ao longo do caminho.

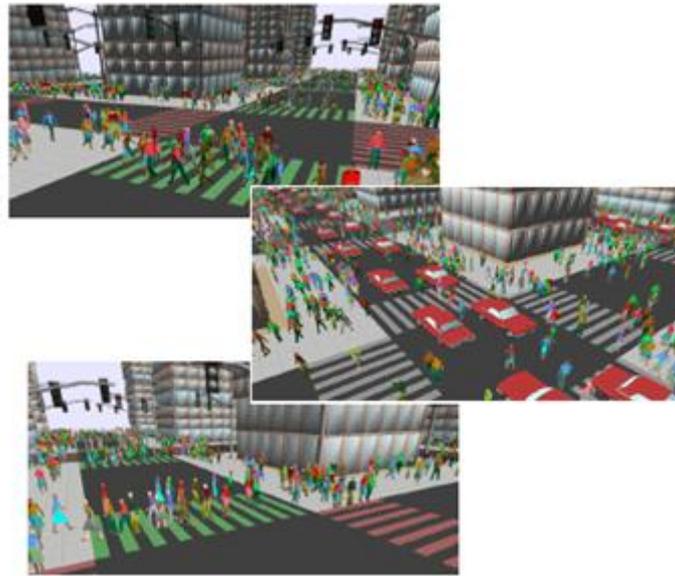


Figura 2.6: Diferentes visões da simulação de uma cidade em *Continuum Crowds*. Extraído de [TEU06].

2.2 *Steering Behaviors*

Steering Behaviors pode ser traduzido como "controle de comportamento" ou "comportamentos guiados". As pesquisas nesta área já possuem uma longa data e vão desde 1940 [NOR48]. De maneira geral, são três as áreas do conhecimento que pesquisam sobre este assunto. São elas:

- robótica;
- inteligência artificial; e
- vida artificial (*artificial life*).

Conforme Reynolds [REY99], o termo *behavior* (comportamento), pode ter vários significados: ações de um humano ou outro animal, ações premeditadas de um simples sistema mecânico, ou complexas de um sistema caótico. Em realidade virtual, isto pode ser sinônimo de "animação". Para fins de estudo, estas entidades são descritas como agentes ou personagens autônomos, e possuem suas representações em um mundo virtual (realidade virtual, por meio de personagens virtuais) ou robôs (que atuam no mundo real). Ainda em [REY99], é proposta uma série de camadas hierárquicas para descrever especificamente o sistema comportamental de locomoção de um agente, conforme pode ser visto na Figura 2.7.

Considerando um agente virtual em seu mundo, com certo nível de autonomia, considera-se que a camada *action selection* (Figura 2.7) é responsável pela tomada de decisão por parte do agente, como por exemplo um comando do tipo "vá até a posição que se encontra a casa vermelha". Já a camada

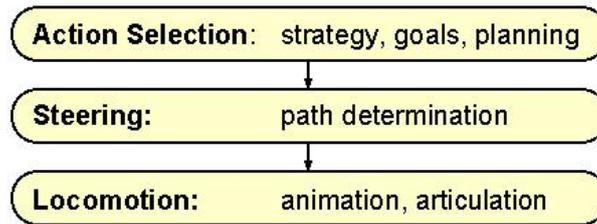


Figura 2.7: Camadas de um sistema de comportamento de um agente autônomo. Extraído de [REY99].

steering representa sub-ações necessárias para chegar até o objetivo. Muitas vezes estas podem ser traduzidas como "contorne a direita, contorne a esquerda, vá rápido, vá devagar". E por último, a camada *locomotion*, é responsável por controlar os níveis de liberdade impostos ao agente, onde são calculadas forças, torques, percepção ou balanço, além das propriedades físicas que atuam no mundo.

2.3 Comportamentos básicos classificados como *Steering Behaviors*

Trata-se de uma série de comportamentos relacionados à locomoção de um agente que podem ser também observados na vida real, quando realizados por humanos ou animais. Em seu trabalho, Reynolds [REY99] aborda os seguintes comportamentos definidos como básicos.

2.3.1 Chegar e evitar (*Seek / Flee*)

Seek é o comportamento de chegar até ao alvo, este comportamento ajusta o agente de maneira que sua velocidade é radialmente alinhada com o alvo. Reynolds ainda destaca que este processo é diferente de uma força de atração (como a gravidade), a qual iria produzir um caminho orbital em torno do alvo. A velocidade desejada é um vetor de direção da posição do agente até o alvo. O tamanho da velocidade desejada pode ser a velocidade máxima, ou ainda a velocidade atual dependendo da aplicação. Sendo assim, o vetor de *steering* é a diferença entre a velocidade desejada e a velocidade atual do agente, conforme equações,

$$\begin{aligned} \text{desiredVelocity} &= \text{maxSpeed} * \text{Normalize}(\text{position} - \text{target}), \\ \text{steering} &= \text{desiredVelocity} - \text{velocity} \end{aligned} \quad (2.1)$$

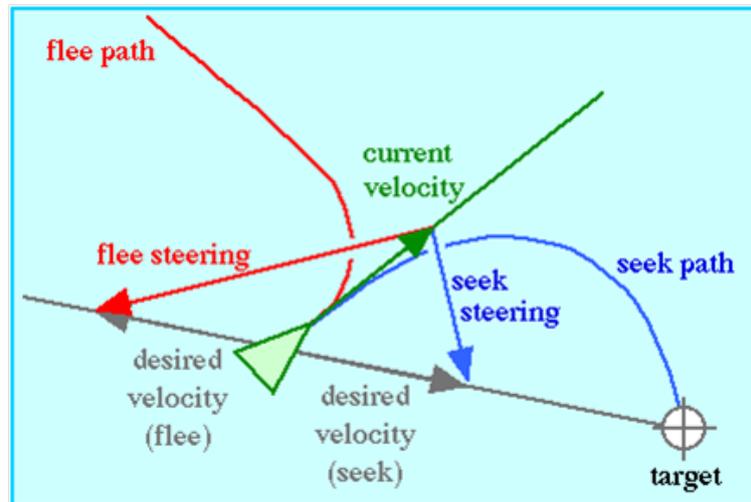


Figura 2.8: Demonstração dos comportamentos *Seek/Flee* proposto por Reynolds. Extraído de [REY99].

Já o comportamento de evitar (*flee*) é exatamente o inverso do chegar (*seek*), e age de maneira que sua velocidade é radialmente alinhada para longe do alvo, assim como a velocidade desejada é apontada para direção inversa. Ambos comportamentos podem ser observados na Figura 2.8. Nesta, observa-se a representação de um vetor de velocidade instantânea (*current velocity*). Já a velocidade desejada (*desired velocity*) é representada para ambos comportamentos *seek/flee* assim como as respectivas forças de *steering* resultantes (*seek steering* e *flee steering*). Como consequência da simulação, observa-se ainda na mesma Figura 2.8, a trajetória que será realizada pelo agente na linha curva denominada *seek path* para o comportamento *seek* e *flee path* para o comportamento *flee*.

2.3.2 Perseguir e fugir (*pursuit/evasion*)

Para Reynolds, ainda em [REY99], o comportamento *pursuit* é similar ao *seek*, mas com o alvo móvel. Para tanto, uma perseguição efetiva requer a predição futura do alvo. Por exemplo, a predição linear da velocidade deve assumir que durante algum tempo o alvo não irá mover-se para alguns dos lados. Frequentemente essa suposição é incorreta, resultando em uma predição que irá ser usada apenas 1/30 por segundo. A posição de um agente T , em uma unidade de tempo no futuro, (supondo que ele não irá manobrar) pode ser obtida escalando sua velocidade em T e adicionando este vetor a sua posição atual. Desta maneira, a perseguição é simplesmente o resultado do comportamento *seek* aplicado a um alvo em uma posição premeditada. De maneira semelhante, o comportamento para fuga é criado fazendo a predição do alvo e aplicando o comportamento de *flee*. Ambos os comportamentos podem ser observados na Figura 2.9, onde um agente, representado pelo triângulo verde, desloca-se em direção a um segundo agente em cinza, no qual, está demonstrado sua posição atual (*now*) e no futuro. (*future*), desta forma, a linha azul indica o comportamento de *pursuit* que será realizado levando em conta a predição futura. De forma inversa ao *pursuit*, é observado o comportamento *evade*

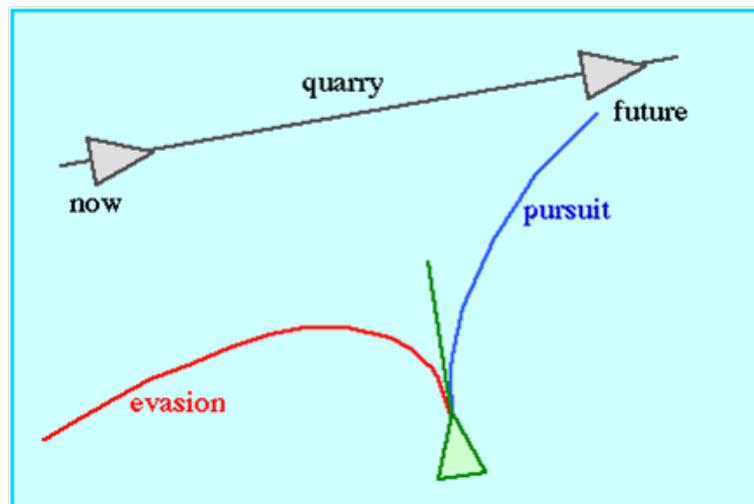


Figura 2.9: Demonstração dos comportamentos *pursuit/evasion* propostos por Reynolds. Extraído de [REY99].

na linha vermelha.

Reynolds, em [REY99], cita que estas predições possuem caráter estritamente básico e salienta que uma técnica ótima para os comportamentos *pursuit/evasion* podem ser encontradas em [ISC65].

2.4 O modelo de canalização de nervuras de plantas

A área de modelagem biológica computacional sempre teve um importante papel na ciência da computação. Nos últimos anos, ganhou força com o aperfeiçoamento das técnicas de render e o desempenho das atuais arquiteturas de computador. Em modelagem biológica, têm-se usado técnicas de visualização da Computação Gráfica como ferramenta para os estudiosos teóricos da morfogênese. Exemplos são as pesquisas apresentadas por [PRU94] da área da botânica, o que inclui modelos de reação/difusão de padrões de pelagem de animais [TUR91], até a pigmentação de conchas do mar [FOW92].

Especificamente, a canalização de nervura de plantas (2.10) usadas em *BioCrowds* é um modelo matemático, biologicamente inspirado na atração que o hormônio auxina exerce sobre o desenvolvimento dos canais, por onde irá fluir o alimento da planta. O modelo que serviu de inspiração para *BioCrowds* pode ser encontrado em [RUN05].

Para o entendimento do embasamento biológico do modelo, é preciso entender o que acontece no mundo real, neste texto, descrito como a hipótese da canalização. Por esta hipótese, a canalização das nervuras em folhas se dá por meio de um hormônio de atração por alimento da planta. Esse hormônio surge espalhadamente pela superfície das folhas, e durante o processo de crescimento da planta, ele atrai as nervuras, ramificando-as no intuito de distribuir o alimento necessário à folha de forma homogênea em sua superfície. Os autores do modelo computacional propõem um método iterativo, que



Figura 2.10: Foto sintética que ilustra, no destaque, o algoritmo inspirado biologicamente para canalização de nervuras. Extraído de [RUN05].

simula o processo de canalização dessas nervuras e suas ramificações conforme a atração exercida pelas auxinas. Este modelo está descrito na Figura 2.11, e comentado a seguir.

Na Figura 2.11, em um primeiro momento, são geradas auxinas (círculos em vermelho), aleatoriamente sobre a superfície da folha virtual 2.11, respeitando certa distância entre elas e as ramificações das nervuras (círculos pretos com centros branco). O próximo passo, e também o mais custoso do processo, é verificar qual auxina está mais perto de qual ramificação das nervuras (Figura 2.11), como ilustrado por meio de retas entre o nó da nervura e a auxina. Consideram-se essas retas como vetores normalizados que são somados para cada nó das nervuras, e então na Figura 2.11 são calculados os vetores resultantes que tiveram influência de auxinas. Desta forma são criados novos nós em direção às auxinas (Figura 2.11). O próximo passo, é excluir as auxinas que estão mais próximos que uma determinada distância das nervuras (Figura 2.11). E, novamente são criadas novas auxinas, respeitando as distâncias com as que sobraram (Figura 2.11), por fim, recalculando a proximidade delas com os nós das nervuras, repetindo assim a iteração. Este processo é síncrono com o de crescimento da folha, sendo assim, completa-se o modelo que simula o crescimento de uma folha e o desenvolvimento de suas nervuras. O resultado do algoritmo pode ser observado na Figura 2.12. O próximo capítulo apresenta detalhes do modelo *BioCrowds*.

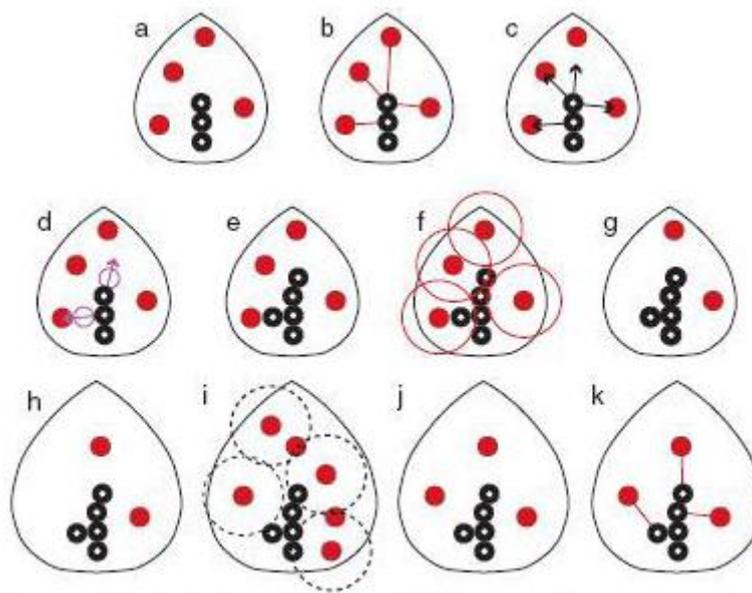


Figura 2.11: Modelo de funcionamento do algoritmo de canalização das nervuras. Extraído de [RUN05].

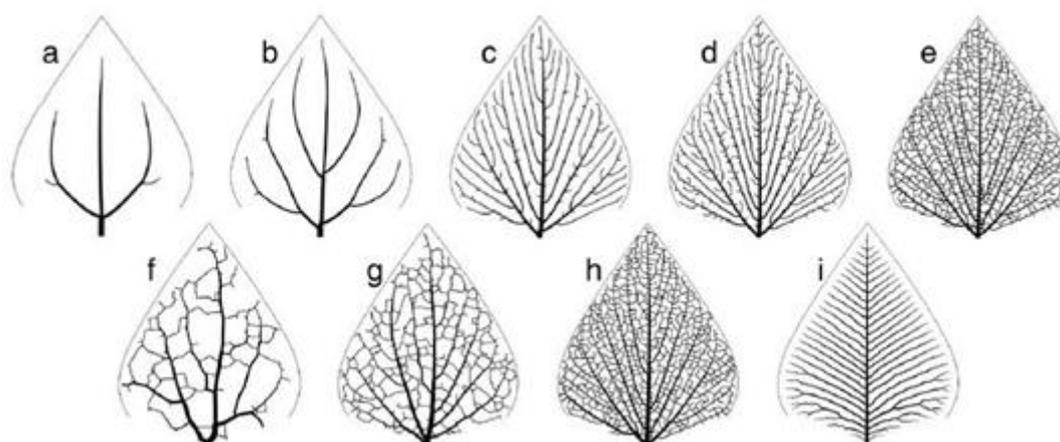


Figura 2.12: Exemplo de seqüências de geração de nervuras nas folhas, simulação com diferentes configurações de parâmetros. Extraído de [RUN05].

Capítulo 3

Fundamentação do modelo *BioCrowds*

Este capítulo apresenta o modelo *BioCrowds* para simulação de multidões desenvolvido no âmbito do laboratório VHLab, com a participação do autor deste trabalho. O modelo foi inspirado no trabalho de Runions et al [RUN05] descrito na seção 2.4, através da adaptação da teoria de competição pelo espaço, observada nas nervuras das plantas, e aplicado em simulação de multidões. Desenvolveu-se o modelo com o principal objetivo de evitar a colisão entre agentes em simulações de altas densidades. Por consequência, emergiram, automaticamente, alguns comportamentos de multidões reais, são eles:

- O surgimento de filas (*lanes*), conforme a teoria do mínimo esforço [STI00], proporcionando uma menor variação angular para agentes que a seguem;
- A redução das velocidades em altas densidades de personagens;
- Geração de efeitos de gargalo em ambientes de espaço restrito.

As próximas seções apresentam detalhes do modelo.

3.1 Adaptação do modelo biológico para simulação de multidões

A idéia principal é representar o espaço vazio, explicitamente usando um conjunto de marcadores, e tratar estes pontos como um recurso pelo qual o agente compete para se movimentar. Um importante aspecto a considerar na dinâmica de população, é o espaço pessoal. Edward Hall [HAL59] propôs o termo *proxemics* para descrever o uso do espaço social. Este espaço é descrito como uma área individual com bordas invisíveis para cada agente. Esta área pode representar a zona de conforto durante a comunicação interpessoal, e pode ser reduzida conforme há aumento de densidade de pessoas em um ambiente. Uma interpretação matemática do *proxemics* de uma pessoa pode ser: uma porção do espaço formada por pontos que estão mais próximos desta pessoa do que qualquer outra. A movimentação de cada agente é baseada nos marcadores que se localizem dentro do seu raio de ação (auxinas do modelo original). Assim cada marcador associado a um agente, influencia na sua movimentação do agente, semelhante ao crescimento da nervura de uma folha.

Embora o método preserve as principais características do modelo biológico de colonização do espaço, importantes modificações foram feitas, são elas:

- A principal diferença entre o desenvolvimento de uma nervura e a movimentação de uma multidão, é a presença de objetivos espaciais na dinâmica da multidão. Nas folhas, as novas nervuras apenas crescem para longe das outras nervuras, e nas multidões, seus movimentos são influenciados pelas intenções individuais ou em grupo;
- No modelo original, o algoritmo para colonização do espaço tem suas nervuras influenciadas igualmente por todas as auxinas. Para a movimentação da multidão, cada marcador tem seu peso individual, baseado na distância do agente ao seu objetivo, bem como a coerência da sua orientação (ângulo entre orientação do agente e objetivo);
- Em contraste com a simulação baseada em auxinas, do padrão de nervuras, os marcadores para animação da multidão persistem: elas são usadas pelos agentes e quando estes se movem para longe delas, são subseqüentemente liberadas para serem usadas por outros agentes.

3.1.1 Descrição do modelo

Em um primeiro momento o mundo virtual é preenchido por marcadores que preencherão as áreas livres para locomoção do agente. Para distribuir esses marcadores foi usado o algoritmo *Dart-Trowing* [COO86]. Sendo assim, obstáculos podem ser definidos, deixando áreas livres de marcadores, como pode ser visto na Figura 3.1, onde as linhas verdes indicam vetores entre a posição do agente até os marcadores que estão o influenciando. Quanto maior o número de marcadores, mais suave é a trajetória, no entanto, isto aumenta o custo computacional de processamento. Devido a esta associação dos marcadores com os agentes mais próximos, quando ocorre a disputa por espaço entre agentes próximos, emerge a partir dos marcadores, uma divisão do espaço entre os agentes que consiste a uma aproximação do Diagrama de Voronoi [OKA08], este é um processo para decomposição do espaço entre elementos em um plano.

Cálculo da direção de movimento dos agentes

A movimentação do agente é simulada iterativamente. Para cada passo da simulação e para cada agente I sua posição $p(t)$, e o vetor $g(t)$, apontando em direção ao destino do agente, são atualizados a cada iteração. Um conjunto S contendo todos os marcadores dentro do espaço pessoal do agente é computado. Este conjunto consiste em todos os marcadores mais próximos do agente em questão do que qualquer outro. O espaço pessoal de um agente, também descrito como *proxemics* [HAL59], pode ser observado na Figura 3.2, bem como os respectivos marcadores pertencentes a cada agente.

Considerando um conjunto de N marcadores $S_I = \{a_1, a_2, \dots, a_N\}$, associado a um agente I , calcula-se um conjunto S' de vetores de orientação para cada N marcador em S .

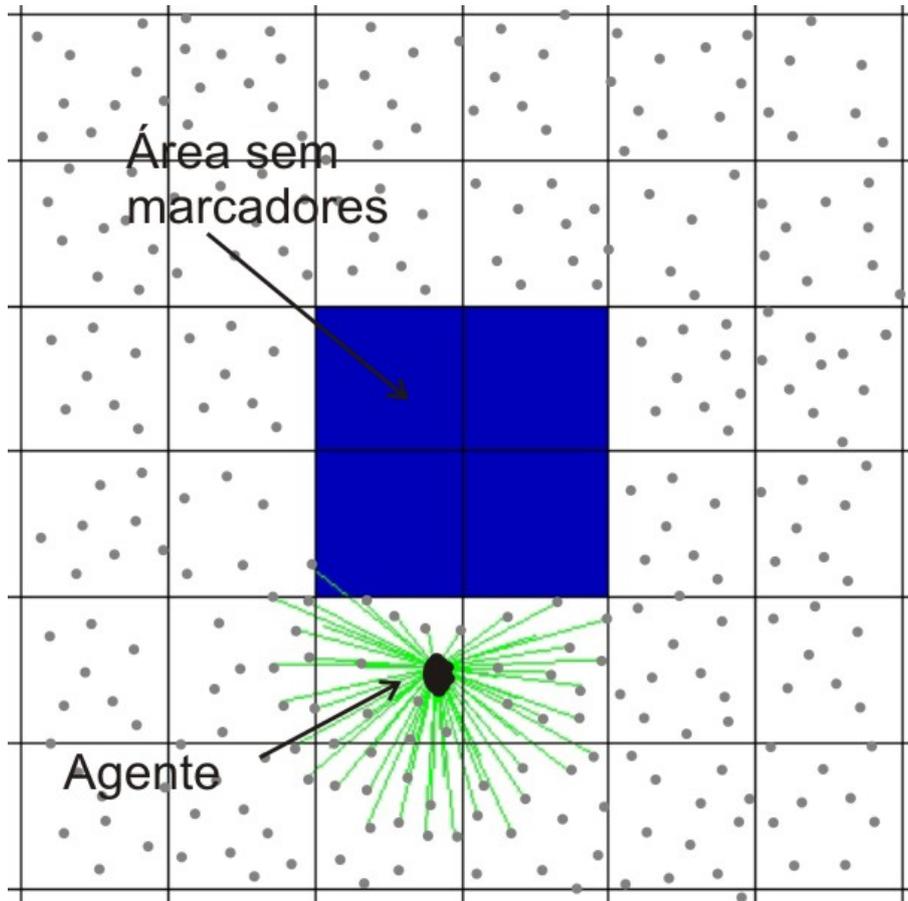


Figura 3.1: Exemplo de agente contornando uma região sem marcadores

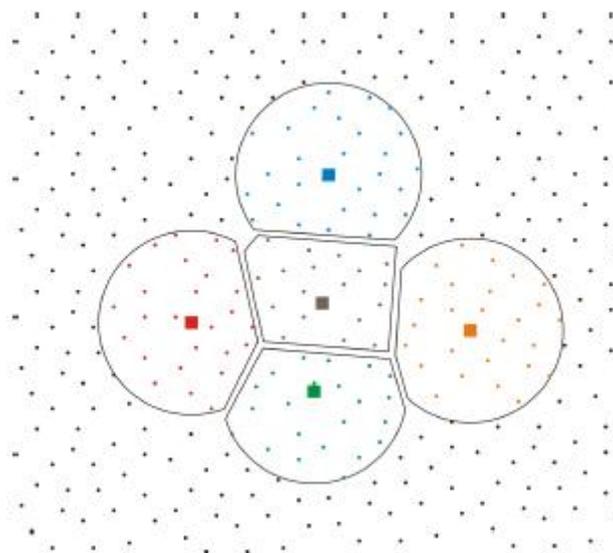


Figura 3.2: Área pessoal (círculo) e marcadores (pontos) associados a cinco agentes (quadrados). Os marcadores capturados por cada agente são mostrados com a mesma cor.

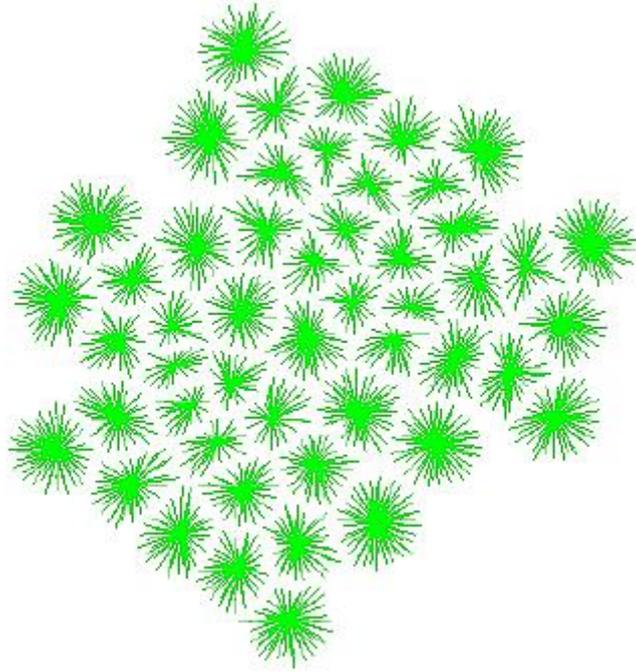


Figura 3.3: Agentes com linhas relacionando estes com seus respectivos marcadores em uma visualização 2D.

$$S' = \{d_1, d_2, \dots, d_N\}, \quad d_k = a_k - p \quad (3.1)$$

Vários destes vetores relacionados aos seus respectivos agentes podem ser observados na figura 3.3. Para que o agente mantenha seu movimento preferencialmente em direção ao seu destino, são calculados diferentes pesos w_k para cada d_k . Estes pesos estão relacionados ao ângulo de d_k com o vetor $(g - p)$, sendo g o destino do agente. Assim, m define a próxima direção de movimento do agente

$$m = \sum_{k=1}^N w_k d_k, \quad (3.2)$$

onde os coeficientes w_k são os pesos calculados usando a equação 3.3

$$w_k = \frac{f(g, d_k)}{\sum_{l=1}^N f(g, d_l)} \quad (3.3)$$

para determinar a função f , assume-se que todos os marcadores a_k , influenciando I , estão na mesma distância $\|d_k\|$ deste agente.

Sendo assim, a função f :

- Alcança o valor máximo quando o ângulo θ entre o objetivo e a orientação for 0;

- Alcança o valor mínimo quando $\theta = 180^\circ$;
- Diminui conforme o ângulo θ aumenta.

Uma possível solução para f é proposta na equação a seguir na equação 3.4:

$$f(g, d_k) = \frac{1 + \cos\theta}{1 + \|d_k\|} = \frac{1}{1 + \|d_k\|} \left(1 + \frac{\langle g, d_k \rangle}{\|g\| \|d_k\|} \right), \quad (3.4)$$

onde $\langle \cdot, \cdot \rangle$ denota o produto interno. Na Figura 3.3 é ilustrada uma forma de visualização do modelo *BioCrowds*, onde existe uma aresta verde ligando cada agente aos respectivos marcadores que os influenciam. O cálculo do deslocamento instantâneo leva em consideração a velocidade máxima (deslocamento por iteração) s_{max} do agente. Calcula-se o deslocamento instantâneo v como:

$$v = s \frac{m}{\|m\|}, \quad \text{onde } s = \min \{ \|m\|, s_{max} \} \quad (3.5)$$

Assim, as posições dos agentes são atualizadas. Nesta seção foram apresentados algumas características básicas do modelo, necessárias para compreensão do próximo capítulo.

Capítulo 4

Steering behaviors e BioCrowds

Após a modelagem de *BioCrowds*, deu-se início o tratamento e desenvolvimento de novas características que permitam o seu uso em diferentes situações de simulação. Sendo assim, novas pesquisas foram definidas para ampliação do modelo *BioCrowds*.

Esta dissertação focou em propor métodos de integração de *BioCrowds* com o conceito proposto por Reynolds, chamado de *steering behaviors*, como já abordado nas seções anteriores.

Em seu estado original o modelo *BioCrowds* provê as seguintes habilidades para um agente autônomo.

- perseguir objetivos;
- controle de velocidade;
- evitar colisão; e
- desvio de obstáculos (uma vez que estes não sejam em pequenas escala e, em um tamanho menor que o raio do espaço pessoal do agente).

Analisando o atual estágio de *BioCrowds*, constata-se que os agentes não possuem capacidade de planejamento de rotas ou estratégias de movimentação, o que compromete gerar *steering behaviors*, como proposto por Reynolds. Além disso, as informações para locomoção são todas obtidas através do mundo, não havendo interação direta entre os agentes. Desta forma desenvolveu-se uma maneira de reproduzir os *steering behaviors* em *BioCrowds*, sem comprometer características inerentes deste em altas densidades de personagens.

Este modelo foi nomeado *Tree Paths* [ROD09], e visa o planejamento de rotas para os agentes. As ações que são realizadas pelo *Tree Paths* são relacionadas à camada superior do modelo conceitual de Reynolds (Figura 2.7), desta forma, gerando *steering behaviors*. A descrição da integração de *Tree Paths* com *BioCrowds* será abordada na próxima seção.

4.1 O modelo *Tree Paths*

Tree Paths é o nome da abordagem proposta neste trabalho e foi inspirada no mesmo algoritmo de canalização de nervuras de plantas, que inspirou o *BioCrowds*, [RUN05], para simular os *steering behaviors*.

A pergunta desta pesquisa visa investigar se o algoritmo de canalização de nervuras pode contemplar a extensão do modelo de *BioCrowds*, a fim de prover *steering behaviors*.

O algoritmo de canalização de nervuras gera segmentos de veias a partir de um ponto inicial (raiz), que iterativamente vai gerando outros a partir de já existentes. Assim, aplicado a trajetória do algoritmo, cria um vasto conjunto de caminhos que partem de um ponto inicial até outros locais, semelhante aos galhos de uma árvore.

No *Tree Paths*, o ponto inicial (raiz) é a posição do agente. Por meio de iterações, que realizam o crescimento das nervuras, segmentos são gerados representando os caminhos que podem ser perseguidos pelo agente, para cumprir sua tarefa. As possíveis tarefas são: chegar até um objetivo, perseguir um alvo ou fugir deste. Na Figura 4.1 pode-se observar uma comparação entre o algoritmo de canalização de nervuras e o *Tree Paths* com suas distintas finalidades, onde são demonstrados os passos iterativos de ambos.

Na próxima seção são apresentadas as adaptações feitas neste modelo de canalização de nervuras, para reprodução de *steering behaviors* em *Tree Paths*.

4.1.1 Adaptação do modelo de canalização de nervuras para o *Tree Paths*

Este modelo, baseado em colonização do espaço, pode ser usado para prover planejamento de rota de um agente em posição fixa ou móvel. Além disto, também é capaz de realizar planejamento de rota para um agente que não tem um determinado objetivo específico, sendo capaz de simular agentes que tentam escapar de algum local ou de outro agente móvel, independente de haver objetivo específico.

Conforme pode ser visto na Figura 4.1, a idéia principal do modelo é criar uma variação do algoritmo de canalização de nervuras, a fim de que os ramos existentes gerem possíveis caminhos, os quais o agente pode optar para chegar até o seu destino. Nesta figura é comparado o processo iterativo do algoritmo de canalização de nervuras com o *Tree Paths*, conforme explicado na seção 4.1.

Algumas modificações foram feitas em relação ao algoritmo de canalização, descrito na seção 2.4.

Próximas, apenas se estas estiverem dentro de um raio de percepção desde ramo, semelhante ao conceito de proxemics em *BioCrowds*. Mas no modelo *Tree Paths*, este raio é chamada de raio de ação, enquanto, no modelo *BioCrowds*, é chamado de espaço pessoal.

- Cada ramificação utiliza as suas auxinas mais próximas, apenas se estas estiverem dentro de um raio de percepção desde ramo, semelhante ao conceito de *proxemics* em *BioCrowds*. Mas, no modelo *Tree Paths*, este raio é chamada raio de ação, enquanto, no modelo *BioCrowds*, é chamado espaço pessoal.

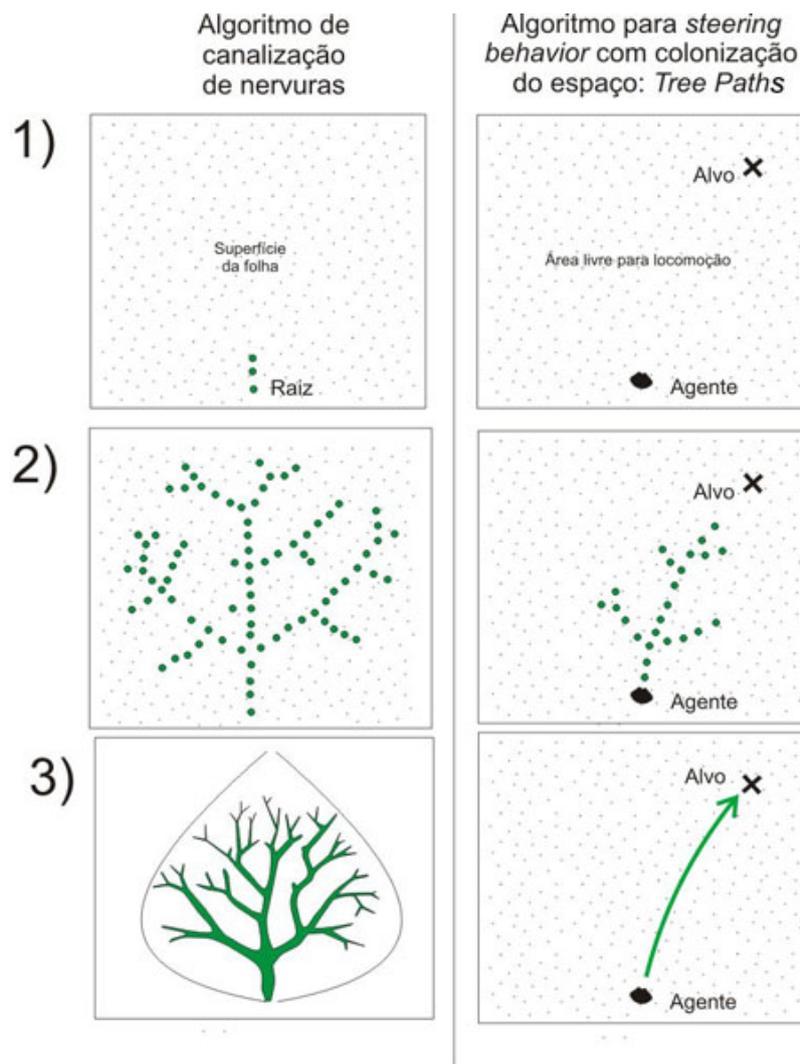


Figura 4.1: Comparação entre o algoritmo de canalização de nervuras e o algoritmo *Tree Paths*. Inicializações, crescimento e os seus devidos resultados.

- Mesmo que um ramo, em uma determinada iteração, possua auxinas associadas a ele, este só irá criar um novo dependendo da probabilidade de se aproximar ao objetivo.
- Diferente do modelo de canalização, onde todas as auxinas geram vetores de influência com o mesmo módulo (ver seção 2.4), em *Tree Paths* os vetores de influência tem sua importância conforme o alinhamento destes com o destino do agente, em um processo semelhante à função de ponderação (ver Eq. 3.3) encontrada em *BioCrowds*. Desta forma, os ramos se desenvolvem tendendo para o objetivo do agente, como pode ser observado na Figura 4.1 à direita.

4.1.2 Definições para o modelo *Tree Paths*

Sendo I um agente, tendo uma posição $p(t)$ para cada iteração t , dependendo do seu comportamento desejado, este pode conter um objetivo fixo ou não. Se um agente tem um objetivo para cada tempo t , é denotado por $g(t)$. Se um agente não tem um objetivo específico, este pode estar apenas tentando escapar de uma determinada região ou seguindo algum agente.

Sendo G , que denota o conjunto de caminhos para um agente, computado usando o algoritmo *Tree Paths*, então G é um conjunto de pontos chave, organizados em um grafo direcional. Para cada iteração, os pontos chave criados são chamadas de nodos, enquanto um segmento une dois nodos.

Outras definições incluem que os caminhos que vão da posição do agente até o seu destino, devido a suas interligações hierárquicas, são representados como uma árvore e podem ser observados na Figura 4.2, onde um agente, representado pela figura do humanóide, tenta chegar ao seu objetivo ilustrado pela bandeira vermelha. Os pontos mais escuros são nodos chaves, onde o agente toma uma decisão de qual caminho seguir. Já os pontos mais claros, representam nodos criados ao longo das iterações, onde a ligação entre eles é um segmento de nodo.

G é computado por meio de duas etapas no algoritmo, dentro de um loop iterativo:

- processamento dos marcadores;
- geração dos novos nodos.

Os marcadores em [RUN05] correspondem ao conjunto de hormônios auxinas das plantas, nas quais, de acordo com a hipótese biológica, emerge o crescimento da folha nas regiões onde não há canais de nervuras. Para o *Tree Paths*, as auxinas são denominadas marcadores, e são usados para definir a direção dos nodos, gerando caminhos como em *BioCrowds*.

Durante cada iteração, um nodo de um caminho é influenciado por todos os marcadores mais próximos deste do que de qualquer outro. Cada nodo n , localizado na posição n , possui um conjunto de N marcadores (localizados na posição m_i) que são mais próximo de n do que qualquer outro nodo. Este conjunto é denotado por

$$S(n) = \{m_1, m_2, \dots, m_N\} \quad (4.1)$$

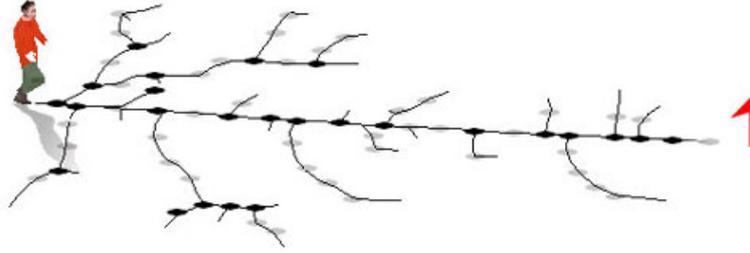


Figura 4.2: O agente tenta chegar ao seu objetivo ilustrado pela bandeira vermelha. Os pontos mais escuros são nodos chaves onde o agente toma uma decisão de qual caminho seguir. Já os pontos mais claros, representam nodos criados ao longo das iterações onde a ligação entre eles é um segmento de nodo.

Por outro lado, uma árvore G^* associada a um outro agente pode usar os mesmos marcadores que G . Apesar de os marcadores serem os mesmos para todos os agentes, o sistema de bloqueio é individual e só são válidos para os nodos de uma mesma árvore. Desta maneira, podem as árvores de diferentes agentes se interceptarem durante o loop iterativo, podendo permitir colisão de trajetórias. No entanto, isto não ocorre devido ao modelo original *BioCrowds*, que garante que os movimentos dos agentes estejam dentro de seus polígonos de Voronoi [OKA08]. Desta maneira o *Tree Paths* se mantém em um caráter mais global de planejamento de rota dos agentes.

Tendo um nodo n , deve ser tomada a decisão se um nodo filho n' irá ser criado a partir de n , ou não. Esta decisão é baseada em uma regra de probabilidade onde compara-se a distância do nodo n_g em relação ao objetivo g (nodo mais próximo do objetivo no tempo t), para com a distância do nodo n . Desta forma, a probabilidade $P_r(n)$, para qualquer outro nodo n , decresce conforme aumenta a sua distância do objetivo:

$$P_r(n) = \frac{\|\mathbf{n} - \mathbf{g}\|}{\|\mathbf{n}_g - \mathbf{g}\|} \quad (4.2)$$

onde $\|\cdot\|$ é norma L^2 de um vetor. Claramente, a probabilidade prioriza a criação de nodos filhos próximos do objetivo. Se um nodo n for gerar um nodo filho n' , este será criado na posição n' por meio de

$$\mathbf{n}' = \mathbf{n} + \alpha \frac{\mathbf{d}(n)}{\|\mathbf{d}(n)\|} \quad (4.3)$$

onde $\frac{\mathbf{d}(n)}{\|\mathbf{d}(n)\|}$ é um vetor unitário representando a direção de crescimento da árvore no nodo n e

α um passo constante que representa o tamanho do segmento do nodo. O vetor $\mathbf{d}(n)$ é baseado nos marcadores em $S(n)$ e em suas coerências com o objetivo \mathbf{g} :

$$\mathbf{d}(n) = \sum_{k=1}^N w_k (\mathbf{m}_k - \mathbf{n}) \quad (4.4)$$

onde

$$w_k = \frac{f(\mathbf{g} - \mathbf{n}, \mathbf{m}_k - \mathbf{n})}{\sum_{k=1}^N f(\mathbf{g} - \mathbf{n}, \mathbf{m}_k - \mathbf{n})}$$

(4.5)

são os pesos de marcadores computados baseado em uma função f não negativa. Esta função prioriza tanto marcadores que estão apontando para o objetivo, como os mais próximos do nodo corrente n . A função f satisfaz as seguintes condições

$$f(\mathbf{x}, \mathbf{y}) = \frac{1 + \cos\theta}{1 + \|\mathbf{y}\|} = \frac{1}{1 + \|\mathbf{y}\|} \left(1 + \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \|\mathbf{y}\|} \right) \quad (4.6)$$

onde o θ é o ângulo entre \mathbf{x} e \mathbf{y} , e $\langle \cdot, \cdot \rangle$ denota o produto interno entre vetores. Observa-se que f decresce conforme aumenta o ângulo entre $\mathbf{g} - \mathbf{n}$ e $\mathbf{m}_k - \mathbf{n}$, e também $\|\mathbf{m}_k - \mathbf{n}\|$. Consequentemente, marcadores alinhados com o objetivo e próximos de \mathbf{n} tem pesos maiores.

Este processo descreve a criação das sequências de nodos conectados por segmentos, mas não a bifurcação na árvore. Para criar bifurcações, um nodo pai n deve estar conectado ao menos com dois diferentes nodos n'_i e n'_j . Quando o primeiro nodo filho, n'_1 , é criado, este retém alguns marcadores em torno de si de acordo com a "distância de restrição", sendo assim o número de marcadores disponíveis de n são reduzidos. Desta forma, $S(n)$ é recomputado com um número reduzido de marcadores, e as Equações (4.2)-(4.3)-(4.4) são reaplicadas para obter o segundo nodo n'_2 , processo este que sempre inclui a probabilidade de criar um novo nodo. Cabe salientar ainda que um nodo só é um "nodo chave" quando possuir mais de um nodo filho, o que representa uma bifurcação.

A Figura 4.3 demonstra a criação de novos nodos no algoritmo *Tree Paths*. O nodo em verde (segmento criado pelos nodos verde e preto) foi criado primeiro, e restringiu alguns marcadores na criação do preto. O nodo azul foi computado com os marcadores que sobraram associados com seu nodo pai n (nodo preto), e este novo nodo também restringe alguns marcadores. O mesmo acontece do nodo preto para o nodo vermelho. Um segundo nodo pode ser criado a partir do mesmo nodo pai (desde que tenham sobrado marcadores). No entanto isto depende da função de probabilidade $P_r(n)$ (Equação 4.2).

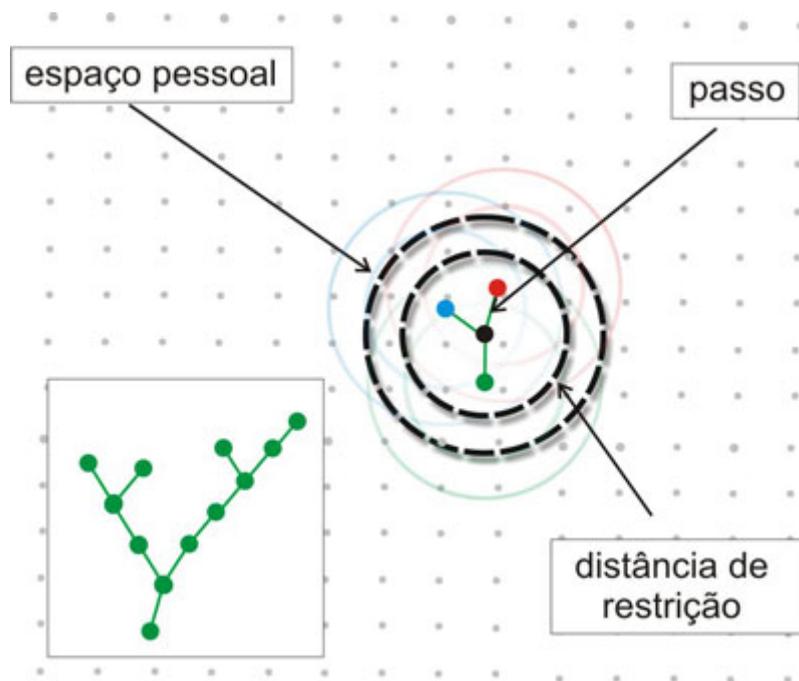


Figura 4.3: Novos nodos são gerados e restringem os marcadores dentro da "região de restrição"

4.2 Integrando *Tree Paths* e *BioCrowds*

O algoritmo *Tree Paths* encarrega-se de estabelecer uma rota na qual o agente deve seguir para realizar o seu objetivo, esta rota é informada ao *BioCrowds* que utiliza nodos em sub-destinos a serem alcançados. Sendo assim, para o *BioCrowds*, cada nodo, é um destino que deve ser alcançado sequencialmente. No entanto, como pode ser observado na descrição do algoritmo *Tree Paths*, quando o algoritmo encontra o caminho até o destino, é gerado um conjunto G de caminhos possíveis, interligados entre nodos chaves. Cabe ao *Tree Paths* repassar para *BioCrowds* qual a posição do nodo seguinte, do melhor caminho a ser trafegado. Dotado desta informação, o *BioCrowds*, que é encarregado da movimentação do agente, define o próximo objetivo do agente, como a posição do próximo nodo n do melhor caminho em G . As interações de *Tree Paths* e *BioCrowds* compartilham o mesmo tempo t , desta maneira a cada passo que a árvore cresce, *BioCrowds* computa um passo do agente, com destino ao nodo seguinte do melhor caminho neste momento. Neste trabalho, optou-se por definir o passo do *Tree Paths* maior que o de *BioCrowds*. Assim, a árvore cresce rapidamente à frente do agente que está usando-a para locomoção. Justifica-se esta decisão para prover a predição do ambiente, pelo *Tree Paths*, antes da locomoção do agente. Quando o agente chega até a um nodo chave (que possui mais de um filho configurando uma bifurcação), é o momento de *Tree Paths* calcular em G , qual dos nodos filhos desta bifurcação deve ser perseguido (melhor caminho). O melhor caminho a ser seguido é computado de uma maneira bastante simples: seja g um caminho de G , $g \in G$, g é o melhor caminho se g contém um nodo n que está mais próximo do alvo do que qualquer outro nodo

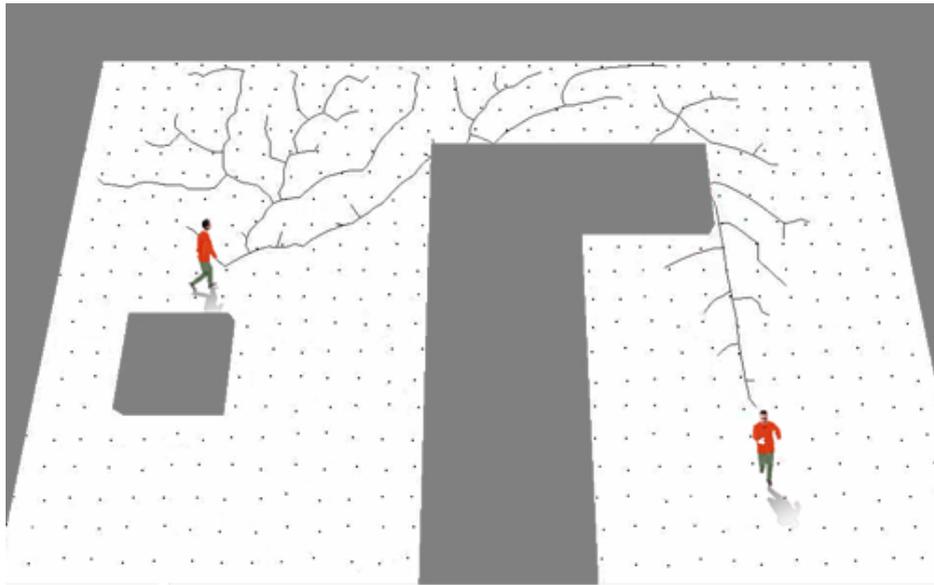


Figura 4.4: Exemplo de aplicação de *Tree Paths* na perseguição de agente. Os marcadores são pontos na superfície. As regiões em preto são regiões sem marcadores. A Figura mostra a árvore desenvolvida de um agente no intuito de chegar até o outro agente.

n pertencente a G . Ainda, algumas considerações são necessárias:

- Toda vez que o agente chega até um nodo n o nodo pai de n é eliminado, e neste processo os marcadores antes bloqueados por este nodo pai são liberados para uso dos próximos nodos que venham a crescer nas seguintes iterações.
- Toda vez que o agente chega a uma bifurcação (nodo chave), e opta por um nodo n que esta associado ao melhor caminho g , além de eliminar o nodo pai deste nodo chave, também são eliminados todos os nodos que não são parentes diretos ou indiretos de n , bem como liberado seus marcadores.
- Toda vez que um nodo n alcança o alvo, a árvore pára de crescer.

Na Figura 4.4 pode ser observado uma aplicação do algoritmo *Tree Paths*. No caso o agente da esquerda, na imagem, persegue o da direita (alvo móvel). Na Figura 4.5 observa-se o algoritmo *Tree Paths* aplicado a uma superfície.

4.3 Perseguição de alvos móveis com *Tree Paths*

Alvos móveis podem representar um problema para *Tree Paths*, uma vez que as ramificações criadas em um tempo t são computadas para alcançar um alvo em uma determinada posição $g(t)$. Conforme o tempo passa, a posição de $g(t)$ pode ser diferente de $g(t + 1)$. Obviamente a mudança da posição do alvo interfere diretamente aos pesos de probabilidade computados em $P_r(n)$ (Eq. 4.2),

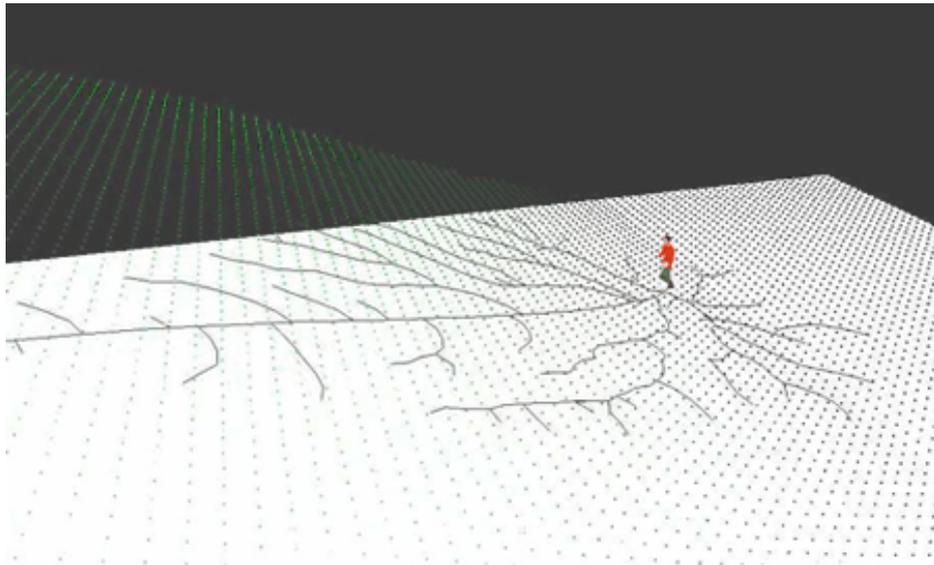


Figura 4.5: Superfície discretizada por meio de marcadores em *Tree Paths*

onde n são os nodos de uma árvore . Portanto, uma pequena adaptação ao modelo se fez necessário conforme segue a seguinte análise . No *Tree Paths* original, enquanto o alvo se movimenta, a árvore cresce em direção ao destino, levando em consideração as probabilidades da equação $P_r(n)$. Conforme o agente toma decisões nas bifurcações, nodos são apagados, com liberação de marcadores. A árvore pára de crescer quando algum nodo encontra o alvo. No entanto, se o alvo se mover novamente, a árvore novamente crescerá em busca do alvo. Deste fato, um problema pode surgir, devido a prioridade dada aos nodos mais próximos do alvo, conforme descrito no esquema da Figura 4.6. No caso, os nodos mais próximos do alvo terão mais chance de crescerem, podendo ocasionar em trajetórias maiores e pouco otimizadas. Assim, a solução adotada é destruir os nodos dentro de um raio de distância a partir da posição do alvo. Esta distância é uma constante denominada distância de ajuste de nodos. Este processo equilibra as probabilidades de crescimento da árvore permitindo que outras ramificações se desenvolvam, como pode ser observado na Figura 4.7

4.4 Colaboração entre agentes

Um comportamento importante é o de cercar o alvo durante a perseguição. Para isto, bastam duas simples modificações no modelo. Conforme dito anteriormente, os marcadores são bloqueados quando fazem parte da árvore de um mesmo agente, ou seja, não podem ser usados para outro nodo de um mesmo agente. No entanto, conforme dito na subseção 4.1.2, os marcadores são liberados para participar do *Tree Paths* de outro agente. Para modelar a colaboração, definiu-se que os marcadores de agentes que perseguem um mesmo alvo não podem ser compartilhados. Assim eles buscam caminhos diferentes para atingir o alvo. Por isso, uma pequena adaptação do modelo proposto para *Tree Paths*

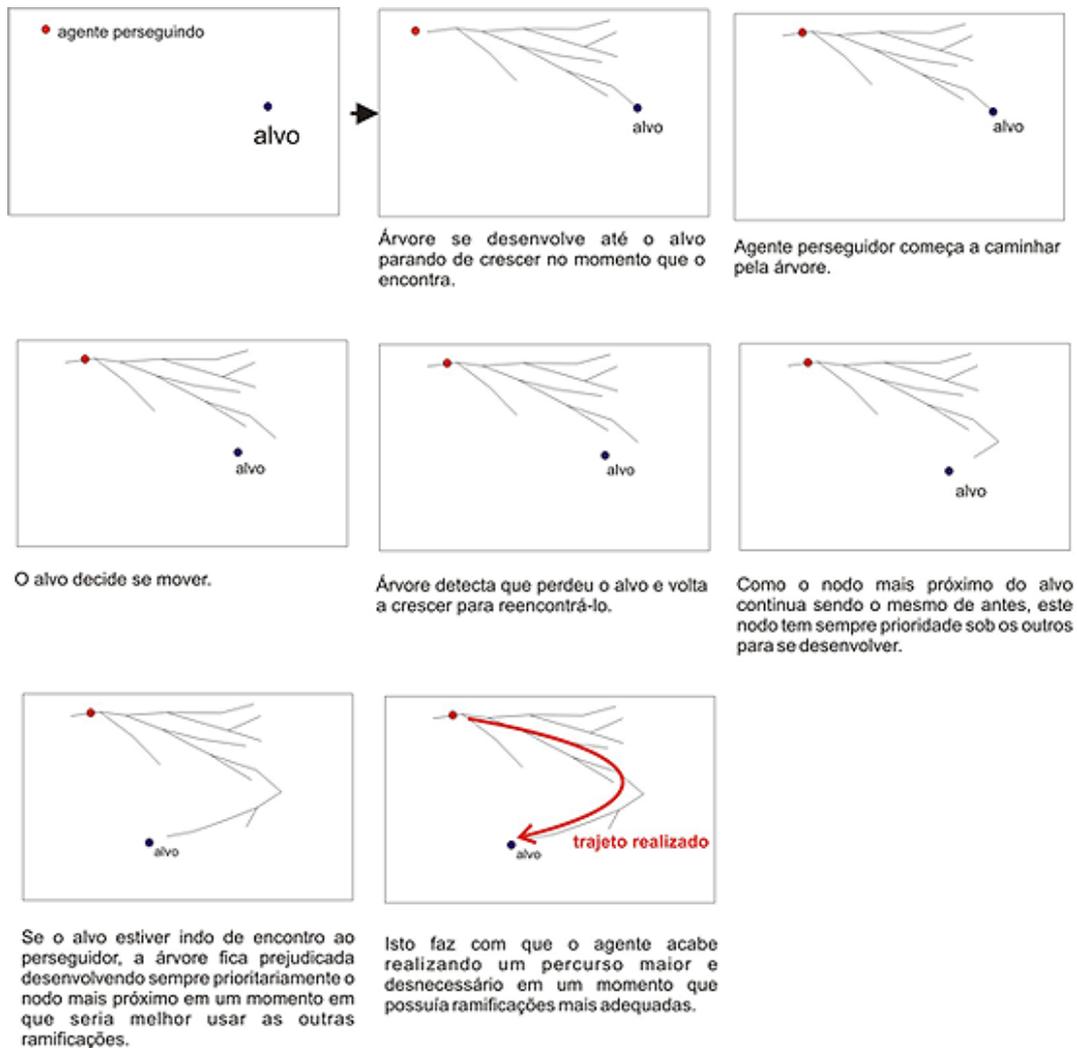


Figura 4.6: O problema da perseguição de alvos móveis

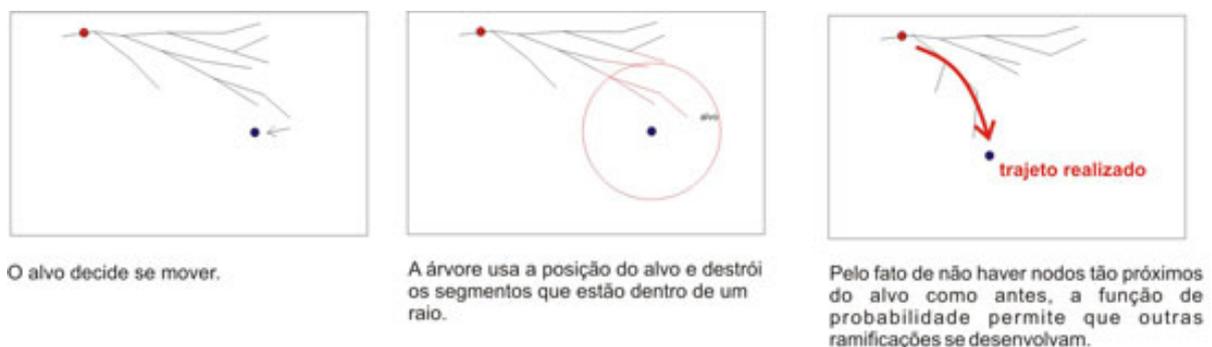


Figura 4.7: Solução para o problema do alvo móvel

permite fazer com que agentes colaborem no intuito de cercar o alvo:

- cada agente colaborativo é apto a eliminar os nodos de outros agentes também colaborativos em um raio de distância, de sua posição atual. Além destes nodos, todos os nodos filhos, diretos ou indiretos destes, são também eliminados. Em uma analogia direta com o conceito de árvore, é realizada uma "poda" nos galhos que interceptam a região da árvore de um agente. Os efeitos dessa ação podem ser observados na Figura 4.8.

Desta maneira, as árvores se desenvolverão tentando evitar sobreposições de suas ramificações, fazendo com que os agentes afastem-se, enquanto vão à direção do alvo.

4.5 Comportamento de fuga

Uma vez que *Tree Paths* é baseado em um algoritmo para colonização do espaço, torna-se fácil criar uma variação que permita a um agente escapar de um determinado lugar sem necessariamente existir um objetivo específico, mas sim, uma área da qual fugir. O algoritmo *Tree Paths* realiza suas iterações em sua forma original, salvo duas modificações necessárias, uma vez que não há mais destino específico:

- não existe mais ponderação em relação ao destino, sendo que os pesos dos marcadores permanecem iguais entre si;
- todos os nodos, que possuem pelo menos um marcador de influência irão dar origem a um novo nodo, não havendo mais cálculo de probabilidade.

O algoritmo *Tree Paths* pára de desenvolver a árvore, quando um dos nodos consegue sair da área que deseja escapar, conforme pode ser observado na Figura 4.9.

4.6 União dos comportamentos colaboração e fuga

Havendo um grupo de agentes que colaboram para perseguir um alvo, estes fazem uso do *Tree Paths*, como descrito na Seção 4.4. Uma vez que este alvo é um agente que tenta não ser pego, ele pode usar a variação do *Tree Paths* descrita na seção 4.5. Para que isto ocorra, é necessário ser determinada a "área de escape", a qual o agente que está em fuga irá usar. Uma proposta para criar essa área é o menor círculo possível (raio), posicionado no mundo, de maneira que englobe todos os agentes perseguidores da simulação e o agente em fuga. Esta área, denotada pelo raio e a posição do círculo no mundo, representa a área da qual o agente em fuga deve fugir. Desta maneira, conforme a seção 4.5, a árvore do *Tree Paths* crescerá até que um nodo tenha escapado dessa região. Para garantir que o caminho gerado pelo *Tree Paths* não leve o agente que foge em direção aos agentes que estão perseguindo-o, são tomadas as seguintes ações:

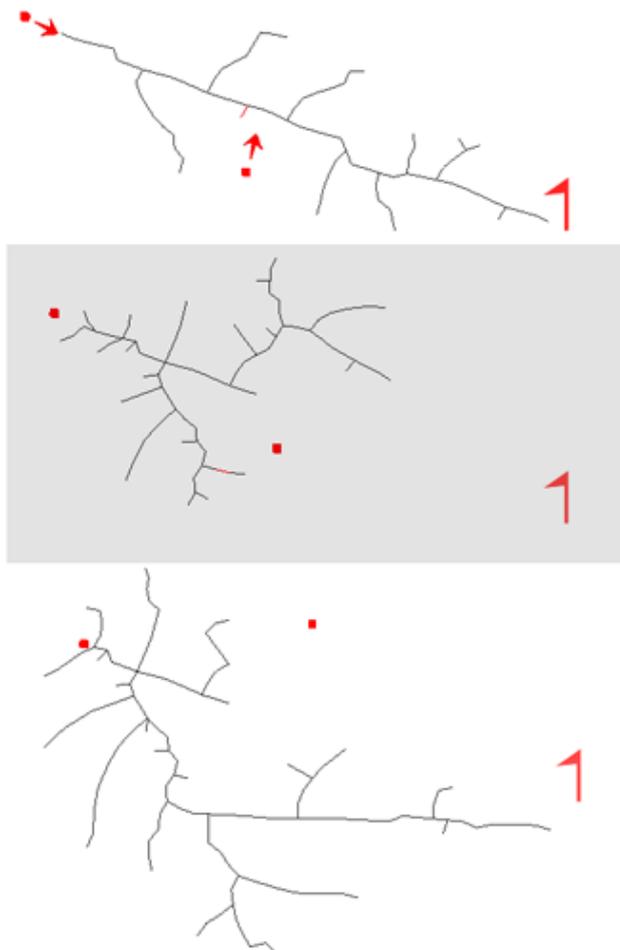


Figura 4.8: Demonstração do que acontece com a simulação quando um agente está habilitado a eliminar os nodos de outro agente, dentro de um raio. O agente mais abaixo no primeiro quadro, conforme se desloca para cima, elimina alguns nodos do agente mais a esquerda que tenta chegar até a bandeira. Nota-se que no decorrer do processo, a árvore do agente volta a crescer por outros ramos.

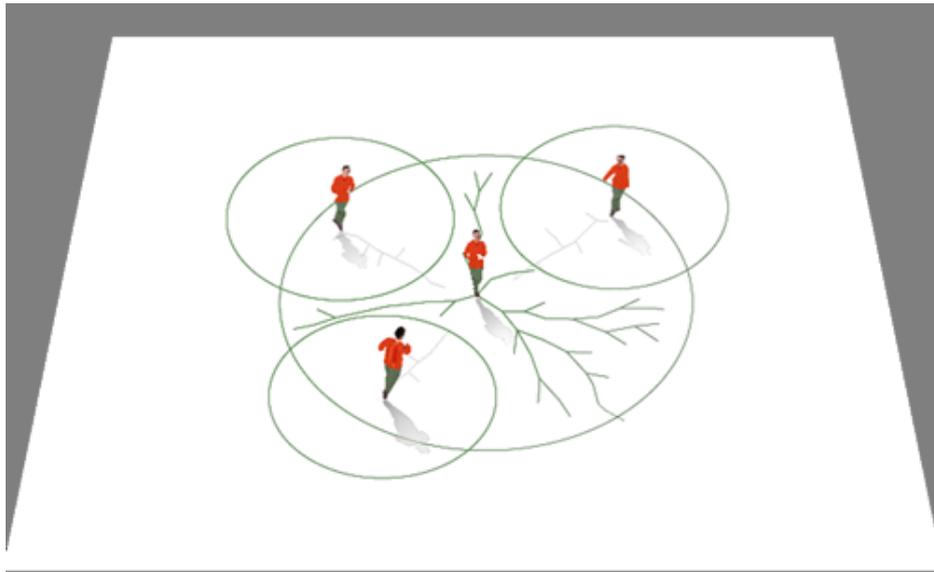


Figura 4.9: Agente (central) fugindo de 3 perseguidores colaborativos.

- os agentes perseguidores bloqueiam seus marcadores não compartilhando com o agente em fuga; e
- os agentes perseguidores, "podam" os nodos da árvore do agente em fuga que estão dentro da "área de escape".

Um exemplo desta dinâmica pode ser observada na Figura 4.9.

Capítulo 5

Comportamentos emergentes de grupos em BioCrowds

Por meio do algoritmo de colonização do espaço, observado nas plantas. Desta forma, *BioCrowds* é capaz de reproduzir comportamentos emergentes como filas (*lanes*) e gargalos (*bottleneck*), entre outros. Um relato mais completo sobre estes comportamentos pode ser encontrado em [STI00].

Neste trabalho investigou-se 2 comportamentos específicos:

- formação geométrica de agentes; e
- alinhamento de agentes.

Estes dois comportamentos possuem grande aplicabilidade em jogos e simulação. Com o intuito de alterar o mínimo possível o *BioCrowds*, apresenta-se a proposta de manipular os pesos associados aos marcadores no espaço, a fim de obter tais comportamentos.

5.1 Formação

Um comportamento importante, a ser controlado, de grupos de personagens, é a formação geométrica, útil em animação e jogos. Por meio deste comportamento, é possível definir áreas, as quais os agentes devem ocupar no tempo, para determinada aplicação gráfica.

Há, ao menos, duas formas diferentes para modelar este comportamento. De forma pré-definida, considerando a geração de posições específicas para cada agente dentro da forma determinada, e posteriormente a distribuição dos agentes neste espaço. Esta abordagem tem pouca flexibilidade, pois exige modificações se a referida área for alterada, ou se mais agentes desejarem participar da animação. Em quaisquer destas situações, seria necessário recalcular cada um dos objetivos, para cada agente.

A segunda abordagem, na qual este modelo se baseia, tem o objetivo de ocupar a área de formação, sem necessidade de calcular objetivos específicos. Para isto, após definir-se uma área, por exemplo, as

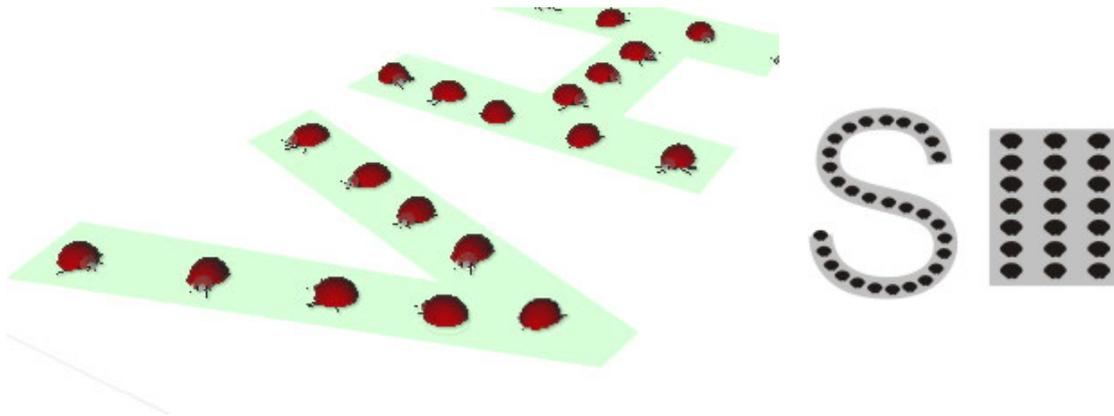


Figura 5.1: Exemplos de formações de agentes em cima de regiões

letras "V" e "H" na Figura 5.1, todos os marcadores no mundo que interceptam com essa região terão seus pesos multiplicados por um fator (nos resultados apresentados no Capítulo 6, foi usado o fator = 10). Desta forma, estes marcadores passam a ter mais importância na movimentação do agente do que os marcadores que estão fora da área.

O algoritmo, para um agente qualquer, baseia-se em três etapas: na primeira, é gerado aleatoriamente um objetivo, localizado dentro da região de formação, e informado para todos os agentes (que compartilham o mesmo objetivo). Na segunda etapa, o *BioCrowds* original é executado, direcionando todos os agentes para a região de formação, até que os agentes passem a utilizar algum marcador de formação (identificados quando a região de formação é especificada). Quando isto ocorre, a terceira etapa se desenvolve. Nesta, a equação 3.3 é substituída por

$$w = \begin{cases} 1 & = \text{se no for marcador de formao} \\ 10 & = \text{se for marcador de formao} \end{cases} \quad (5.1)$$

onde os pesos dos marcadores não se diferenciam em função da posição e orientação do objetivo, mas sim, se são marcadores de formação ($w = 10$) ou não ($w = 1$).

Assim, ao receber estímulos de marcadores de formação, os agentes competem pelo espaço de formação, devido ao peso maior destes marcadores em relação aos que estão fora da área. Consequentemente, os agentes acabam por se manter nas áreas de formação. Na Figura 5.2, observa-se uma sequência de agentes que vão de encontro a uma área de formação. Quando estes a atingem, a competição do espaço, inerente de *BioCrowds*, faz com que os agentes se distribuam pela área.

5.2 Alinhamento entre agentes

Este comportamento pode ser usado juntamente com o comportamento de formação. Basicamente ele visa prover o alinhamento dos agentes em quatro direções possíveis, como pode ser observado na

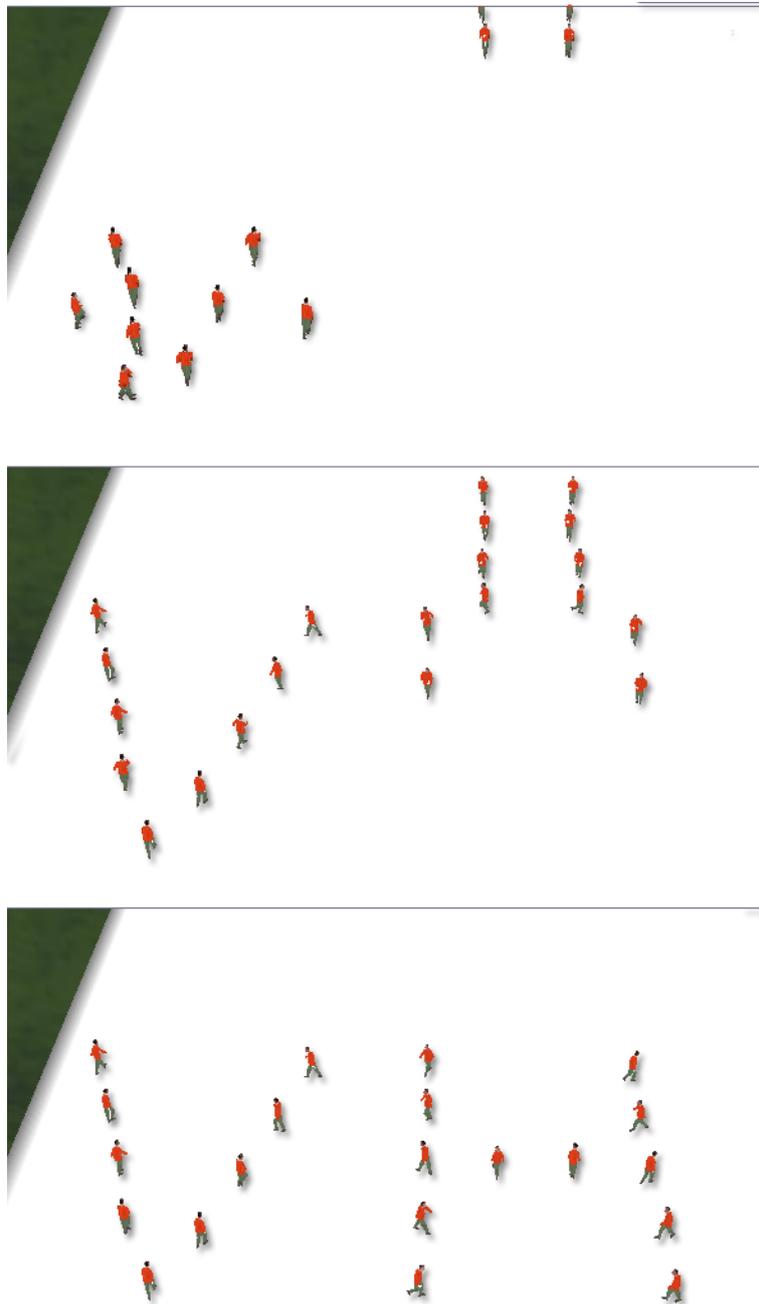


Figura 5.2: Agentes chegando em uma área de formação que ilustram "VH".

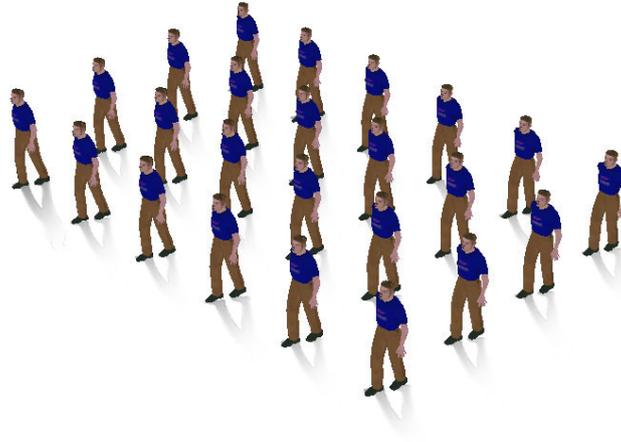


Figura 5.3: Alinhamento entre agentes.

Figura 5.3, ou apenas em duas, semelhante ao que seria uma fila.

Igualmente ao comportamento de formação, o alinhamento pode ser usado para várias aplicações na área de animação. Este comportamento, da mesma maneira que o anterior, necessita de uma simples modificação na maneira de ponderar os marcadores para influenciar os movimentos dos agentes, em *BioCrowds*.

As possíveis direções são horizontal e vertical, conforme mostra a Figura 5.4, no topo. O algoritmo visa, primeiramente identificar quais são os marcadores localizados dentro da região da máscara de formação. Para estes, seu peso aumenta de acordo com a equação $w_k = distance_{i,k}$ onde a distância é calculada do agente i para cada marcador k . Além disso, os pesos dos marcadores são acumulados em uma única estrutura de dados, fazendo com que diferentes agentes contribuam com o peso de um mesmo marcador. No entanto, cada marcador irá influenciar apenas um agente (o mais próximo), porém utilizando o valor acumulado.

A Figura 5.3 ilustra este comportamento e na Figura 5.5 uma visão esquemática superior, demonstrando as regiões de peso mais alto da máscara.

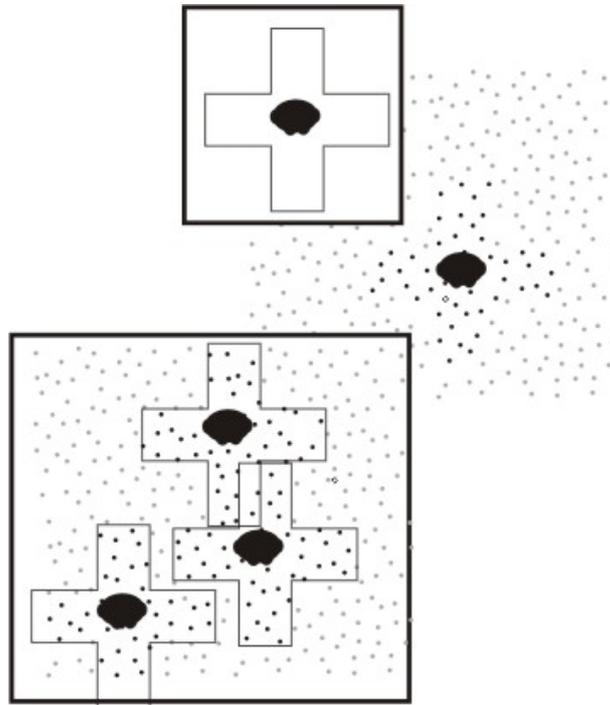


Figura 5.4: Exemplo de máscara de pesos, que fará os agentes colaborarem para alinhar-se na horizontal e vertical.

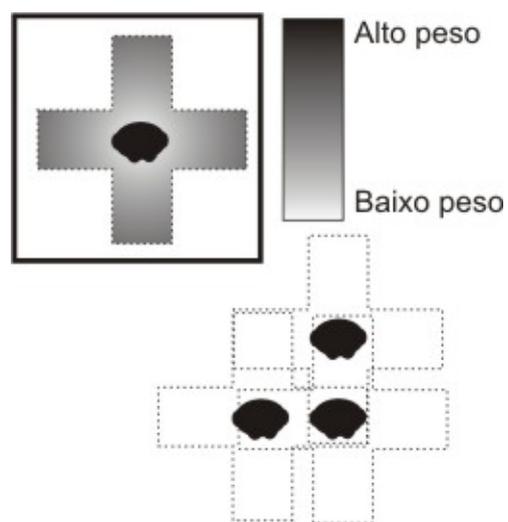


Figura 5.5: Resultado da simulação após agentes se alinharem.

Capítulo 6

Resultados

Os resultados apresentados neste capítulo foram alcançados por meio de um protótipo desenvolvido para este trabalho em C++/OpenGL. Desenvolveu-se um ambiente virtual 3D onde é possível criar agentes interativamente, definir *behaviors* podendo usar os algoritmos de *BioCrowds* e *Tree Paths*, além de também poder setar regras para formação e alinhamento.

De forma interativa, pode-se setar diferentes destinos para os agentes, além de definir comportamentos de perseguição, fuga, definir áreas para formação e definir obstáculos. Todas as ações podem ser realizadas com o mouse interativamente. O protótipo ainda inclui opções como salvar marcadores, a fim de não necessitar o reprocessamento destes toda vez que o protótipo é iniciado, gravação de vídeo, também inclui, geração de arquivo com registros de trajetórias feitas pelos agentes em formato compatível com o usado nas ferramentas de visualização no âmbito do laboratório VHLab e carregar arquivos de imagem como mapa para criação de áreas de formação.

6.1 Considerações sobre parâmetros usados

Todas as simulações foram feitas em uma área que representa 46 x 32 metros do mundo real. Além disso, foi configurada para uma taxa de visualização de 30 FPS (*frames per second*), onde cada iteração representa 1/30 segundos do mundo real.

Infelizmente, *Tree Paths* ainda precisa que seus parâmetros sejam ajustados para respeitar os obstáculos do ambiente. Por exemplo, na Figura 6.1, *Tree Paths* pode gerar caminhos que transponham o obstáculo devido ao raio de ação dos nodos receber influência de marcadores que estariam do outro lado do obstáculo.

Isto acontece porque, tanto *Tree Paths* quanto o *BioCrowds*, discretizam o ambiente em marcadores, sendo que estes representam porções povoáveis do mundo. Desta forma, os resultados das simulações são dependentes de como os marcadores são dispostos no ambiente. As simulações feitas nesta dissertação, reproduzem um ambiente em escala real, onde os agentes possuem a escala e restrições físicas de uma pessoa real. Os parâmetros de *BioCrowds*, para este tipo de simulação já

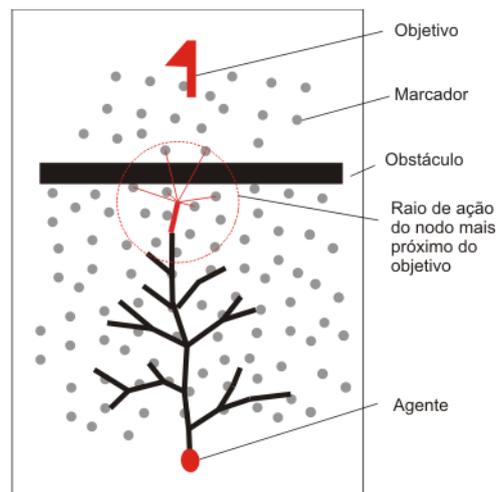


Figura 6.1: Problema em *Tree Paths* com marcadores ocluídos por obstáculos.

foram estudados, e uma boa calibração destes é apresentada nas próximas seções. Já no *Tree Paths*, a calibração de parâmetros usadas foi:

- o raio de ação dos nodos deve ser maior que a distância mínima entre os marcadores;
- a distância para restrição dos marcadores deve ser menor que o raio de ação;
- o tamanho de um segmento de nodo deve ser maior que o módulo do vetor que representa a velocidade máxima em *BioCrowds*;
- os obstáculos do ambiente que representam regiões não passáveis, devem ser mais largos e compridos do que a soma do raio de ação dos nodos e o tamanho do segmento destes.

Baseado nestas observações, e simulando ambientes que possuem obstáculos mais largos e compridos do que a soma do raio de ação dos nodos e o tamanho do segmento destes, os parâmetros adotados em *Tree Paths*, são relacionados nas próximas seções. Para a distribuição dos marcadores, foram usados métodos distintos em ambos modelos, o *BioCrowds* foi usado o algoritmo *Dart-Trowing* [COO86], com mais de setenta por cento da superfície preenchida, ao comparado a uma situação de cem por cento, onde haveria uma distribuição uniforme respeitando a distância mínima (0.2 m) entre os marcadores. Já os marcadores para os nodos do *Tree Paths* foram distribuídos de uma forma linear, respeitando a distância mínima (1.5 m) entre eles. No final deste processo, foi inserido um ruído de $-0.100m$ até $0.100m$ nas posições de cada marcador, no intuito de evitar que a uniformidade entre eles pudesse criar alguma forma de equilíbrio nos equacionamentos de *Tree Paths*. Uma Comparação entre as densidades de marcadores usadas em *BioCrowds* e *Tree Paths* de forma proporcional, pode ser observada na Figura 6.2. *BioCrowds* possui uma média em marcadores por metro quadrado de $17.5m^2$, enquanto *Tree Paths* apresenta $0.4m^2$.

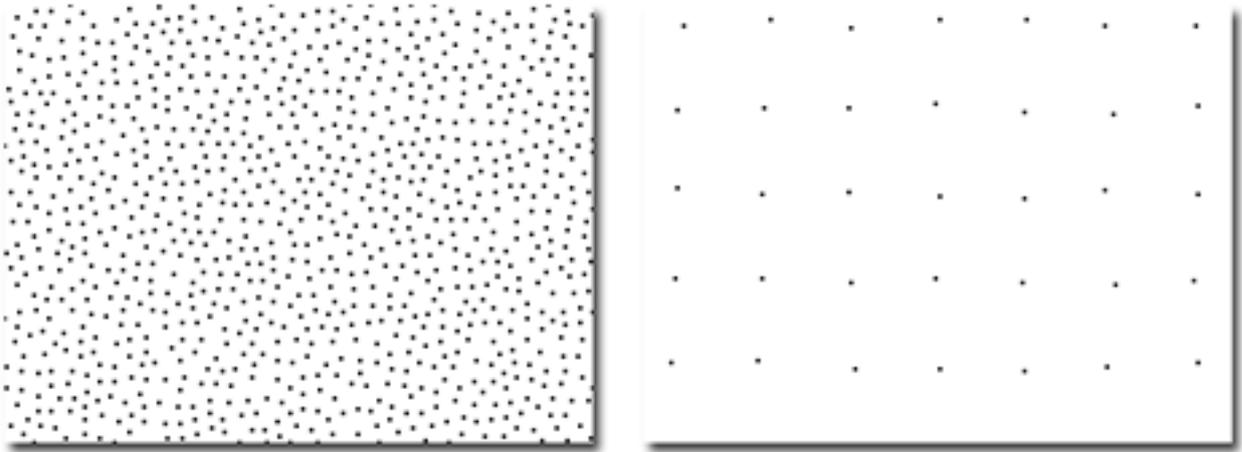


Figura 6.2: Comparação proporcional da média de marcadores por metro quadrado em *BioCrowds* (à esquerda sendo $17.5m^2$) e *Tree Paths* (à direita sendo $0.4m^2$) usadas nas simulações.

- Parâmetros usados em *BioCrowds*

raio de proxemics = 1.25 m

distância mínima entre marcadores = 0.2 m

tamanho do passo máximo de um agente = 0.4 m

velocidade máxima de um agente = 1.8 m/s

- Parâmetros usados em *Tree Paths*

Raio de ação = 2.75 m

Distância para restrição dos marcadores = 1.75 m

Distância linear entre nodos (tamanho do segmento) = 0.6 m

Distância mínima entre marcadores = 1.5 m

Distância de ajuste de nodos = 10 m

Distância para corte de nodos para árvores para agentes colaborativos ou em fuga = 2.5m

- Parâmetros para formação

Fator de escala para vetores de influência relacionados ao marcadores em área de formação

6.2 Simulações realizadas

Apresenta-se uma avaliação que ilustra o planejamento de rota de *Tree Paths* com os parâmetros utilizados. Isto é comparado a outro algoritmo, que objetiva gerar o melhor caminho entre um ponto e outro. Para isto, um agente que percorreu duas vezes a trajetória de uma posição inicial até um objetivo, em um mesmo ambiente. Na primeira simulação, foi usado o algoritmo A-Estrela [LOG98], atuando em uma discretização do ambiente em células de $1m^2$. Na segunda, foi usado o *Tree Paths* conforme as configurações básicas da seção 4.1.2. O resultado pode ser observado na Figura 6.3, onde o agente percorreu uma distância de $78.05 m$ enquanto com o *Tree Paths* a distância percorrida é de $83.60 m$. Percebe-se que, apesar da distância percorrida ser maior, *Tree Paths* (não é objetivo prover o menor caminho), este poderia ser re-inicializado em qualquer nodo da árvore.

As simulações apresentadas nesta seção estão relacionadas na Tabela 6.2, e foram baseadas na tentativa de reproduzir steering behaviors entre agentes, constituindo um resumo destes comportamentos. Desta forma, os comportamentos avaliados foram planejamento de rota, perseguição de alvo móvel, perseguição colaborativa de alvo móvel, fuga de uma área, perseguição e fuga, formação e alinhamento.

- 1. A simulação demonstrando o planejamento de rota tem o objetivo de destacar como o algoritmo *Tree Paths* interage com os marcadores desenvolvendo a árvore em direção ao objetivo do agente, conforme ilustrado na Figura 6.5. Nesta simulação, observa-se um agente que tenta chegar até um objetivo representado pela bandeira vermelha, os retângulos pretos representam obstáculos (região sem marcadores), e os marcadores são os pequenos pontos.
- 2. Perseguição de alvo móvel, demonstra o processo de correção da árvore relatado na seção 4.3, onde um agente persegue o outro que vaga pelo ambiente. Esta simulação é observada na Figura 6.4 onde o agente da esquerda persegue o agente da direita. Nota-se que do segundo para o terceiro quadro, há uma mudança considerável na configuração da árvore, que faz com que o agente o agente circunde o obstáculo pelo outro lado. Isto foi possível devido ao ajuste dos nodos que ocorreu nas iterações entre estes quadros: a eliminação de alguns nodos mais próximos do alvo permitiu que os nodos que seguiam para baixo crescessem.
- 3. A colaboração para perseguição de alvo móvel pode ser observado na Figura 6.6, onde 2 agentes colaborativos à esquerda contornam o obstáculo para chegar até o agente que vaga pelo cenário. Observa-se que as árvores de ambos não se interceptam conforme especificações da Seção [sub:Colaboração-entre-agentes].
- 4. O comportamento de fuga demonstra um agente tentando sair do cerco de vários agentes que permanecem parados (agentes estáticos), e pode ser observado na Figura 6.7, onde a área do círculo vermelho representa o círculo do qual este deseja escapar. Como descrito na Seção 4.5, esta área deve circundar todos os agentes envolvidos na simulação, desta forma, o primeiro

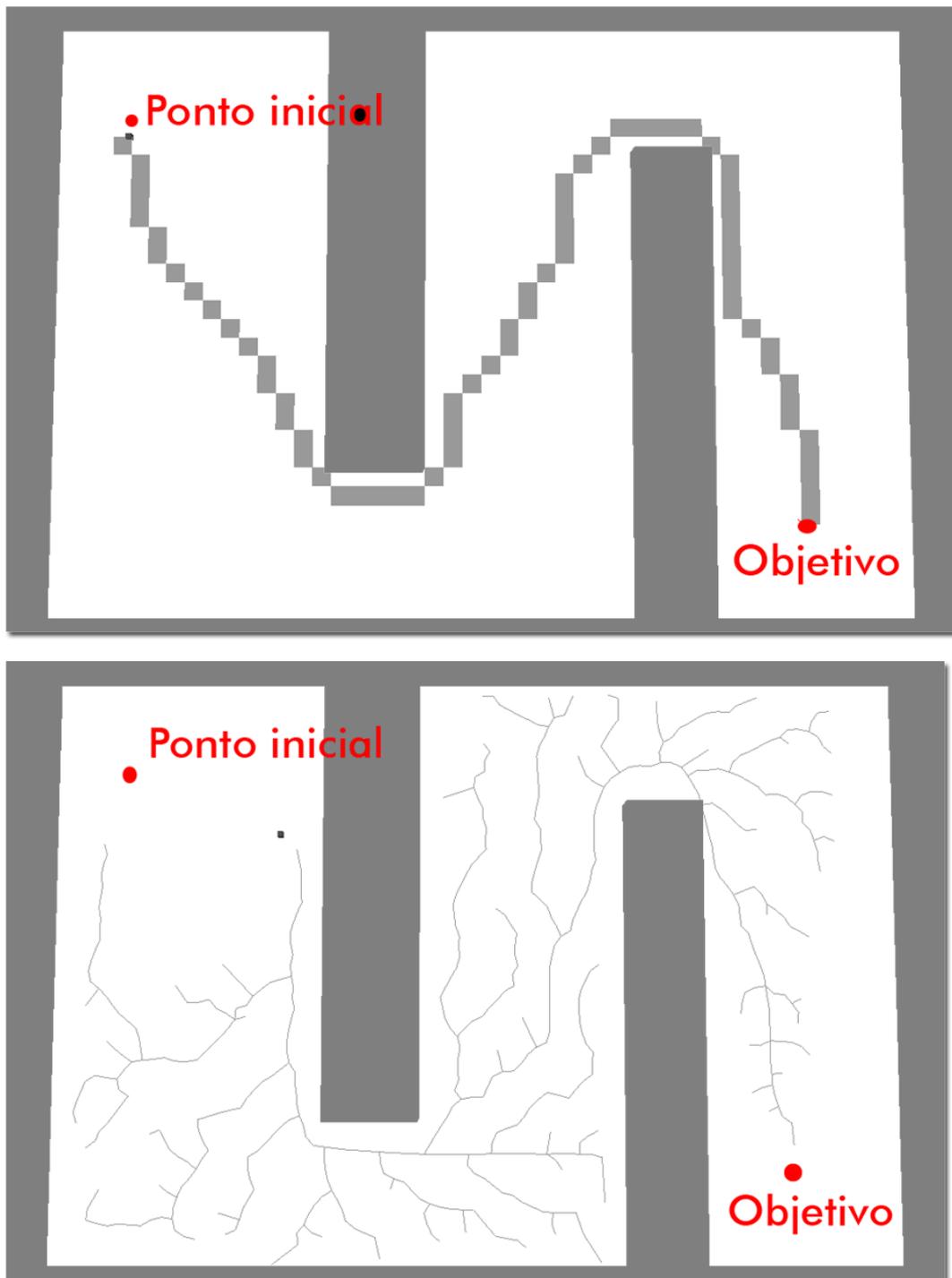


Figura 6.3: Comparação da distância percorrida entre A-Estrela (acima) e *Tree Paths* (abaixo). Na primeira simulação foi percorrida uma distância de 78.05 m, e na segunda, o resultado foi 83.60 m percorridos.

Tabela 6.1: Configurações utilizadas para simulação dos comportamentos.

| Comportamentos | Agentes envolvidos | Configurações principais em <i>Tree Paths</i> | Obstáculos |
|---|--|--|------------|
| Planejamento de rotas. | 1 agente | Configuração básica (Seção 4.1.2). | Sim |
| Perseguição de alvo móvel. | 1 agente persegue outro agente que vaga sem destino. | Agente perseguidor realiza ajuste dos nodos ao perder o alvo de sua árvore (Seção 4.3) . | Sim |
| Perseguição colaborativa de alvo móvel. | 2 agentes perseguem outro agente que vaga sem destino | Agentes perseguidores bloqueiam marcadores (Seção 4.4), e realizam ajuste dos nodos quando perdem o alvo. | Sim |
| Fuga de uma área. | 1 agente tenta escapar do cerco de 4 agentes estáticos | Agente em fuga usa as configurações da seção (4.5) agentes estáticos podam a árvore do em fuga e bloqueiam marcadores para este também. | Sim |
| Perseguição colaborativa e fuga. | 1 agente em fuga de 4 agentes que tentam pegá-lo. | Agente em fuga usa as configurações da Seção (4.5). Agentes perseguidores podam a árvore do agente em fuga e bloqueiam marcadores (Seção 4.6). | Sim |
| Formação. | 21 agentes. | Configurações para preenchimento de áreas (Seção 5.1). | Não |
| Alinhamento | 51 agentes | Configurações para alinhamento de agentes (Seção 5.2) | Não |

nodo da árvore que conseguir sair deste círculo representa o melhor caminho a ser seguido. Observa-se ainda que os marcadores em torno dos agentes estáticos, são bloqueados fazendo com que a árvore não intercepte com eles.

- 5. Na Figura 6.8 é observado a fuga de um agente (à direita) enquanto os outros (à esquerda) colaboram a fim de pegá-lo. Observa-se que as árvores dos agentes colaborativos não se interceptam, no entanto, devido a restrição do ambiente, estes agentes acabam colonizando rapidamente todo o espaço, e assim, prejudicando o processo de colaboração. Já agente em fuga, que possui sua área para escapar representada pelo círculo vermelho, acaba por atravessar o cenário da direita para a esquerda devido a procura por áreas livres.
- 6. Na Figura 6.9, é ilustrado a extensão de *BioCrowds* para formação, onde os marcadores em vermelho interceptam com uma área de formação em forma de estrela. Observa-se que os agentes tem como o destino o centro da estrela, mas ao entrar em contato com os marcadores de formação, passam a usar o equacionamento definido na Seção 5.1.
- 7. Na Figura 6.10 é demonstrado a extensão de *BioCrowds* para alinhamento entre agentes usando a máscara de pesos para alinhamento vertical e horizontal, onde a sequência de quadros demonstra o progresso do alinhamento, devido a influência das máscaras de pesos. Observa-se que o alinhamento horizontal e vertical, com a máscara de pesos proposta na Seção 5.2, pode acarretar algumas falhas em alguns pontos da formação. Para conseguir um alinhamento perfeito horizontal e vertical, seria necessário uma distribuição linear dos marcadores de *BioCrowds*.

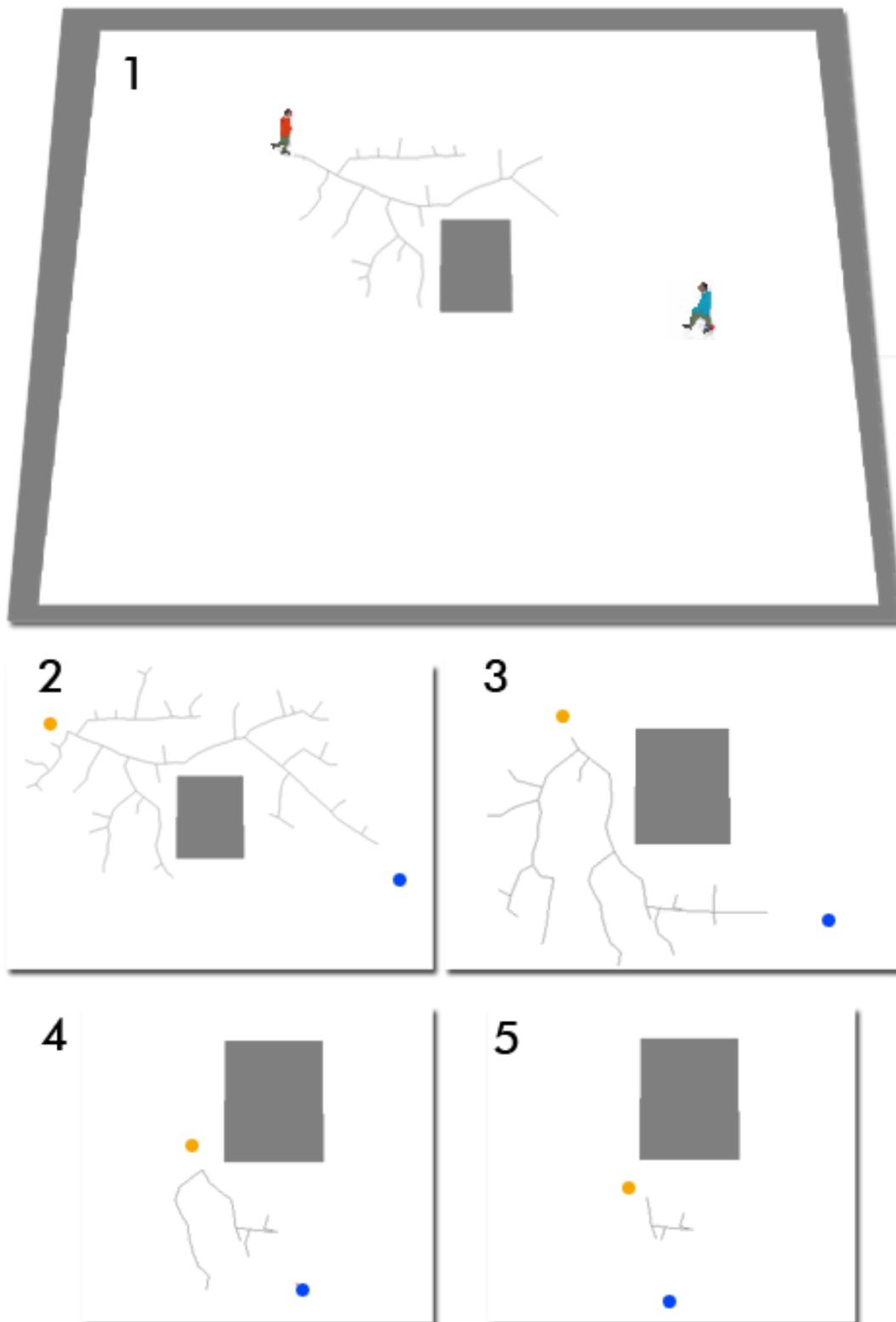


Figura 6.4: Perseguição de alvo móvel. O agente à esquerda persegue o agente à direita que vaga pelo ambiente.

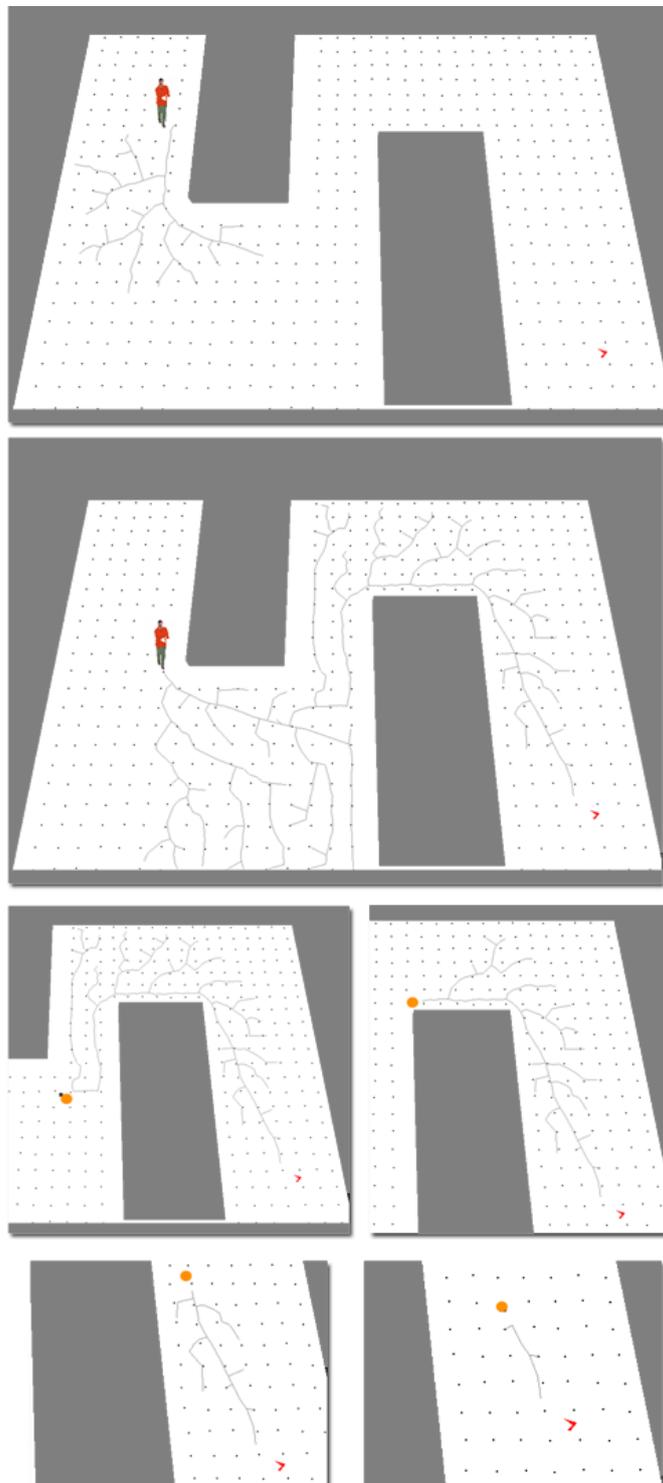


Figura 6.5: Planejamento de rota em *Tree Paths*. O agente é o cubo em preto, os obstáculos são representados como retângulos.

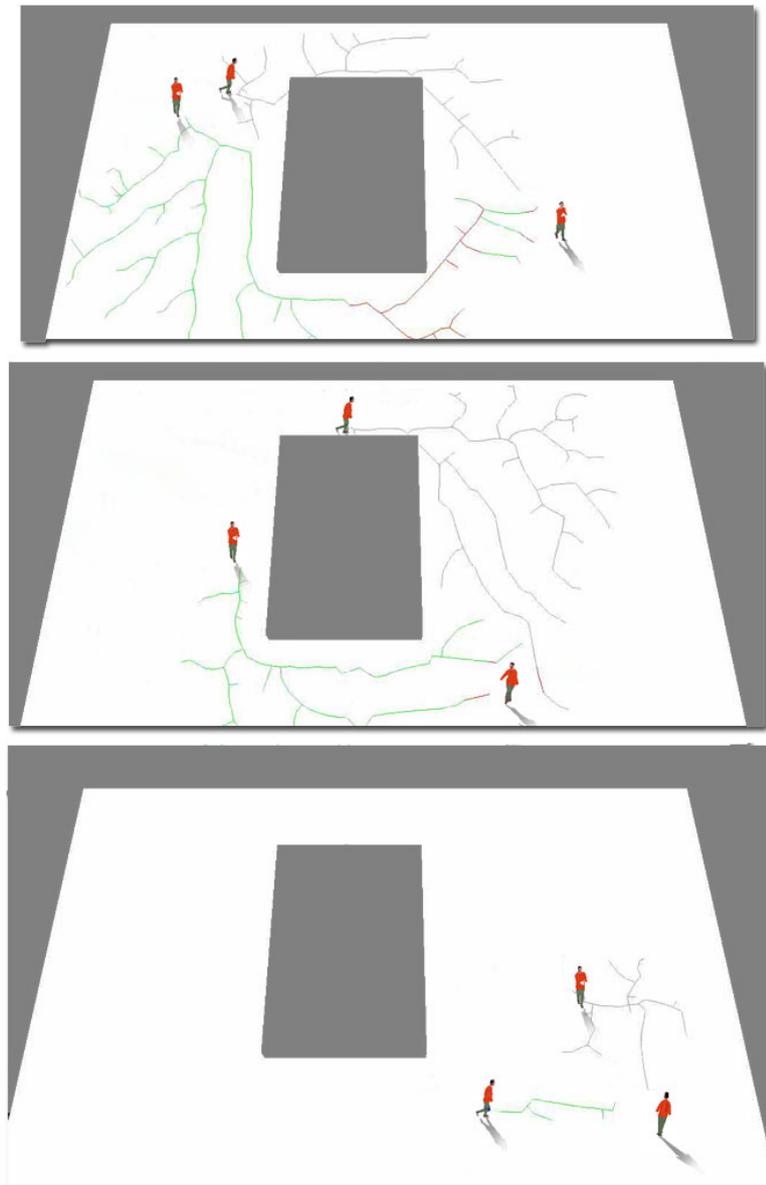


Figura 6.6: Perseguição colaborativa de alvo móvel. Os agentes à esquerda colaboram para alcançar o agente à direita.

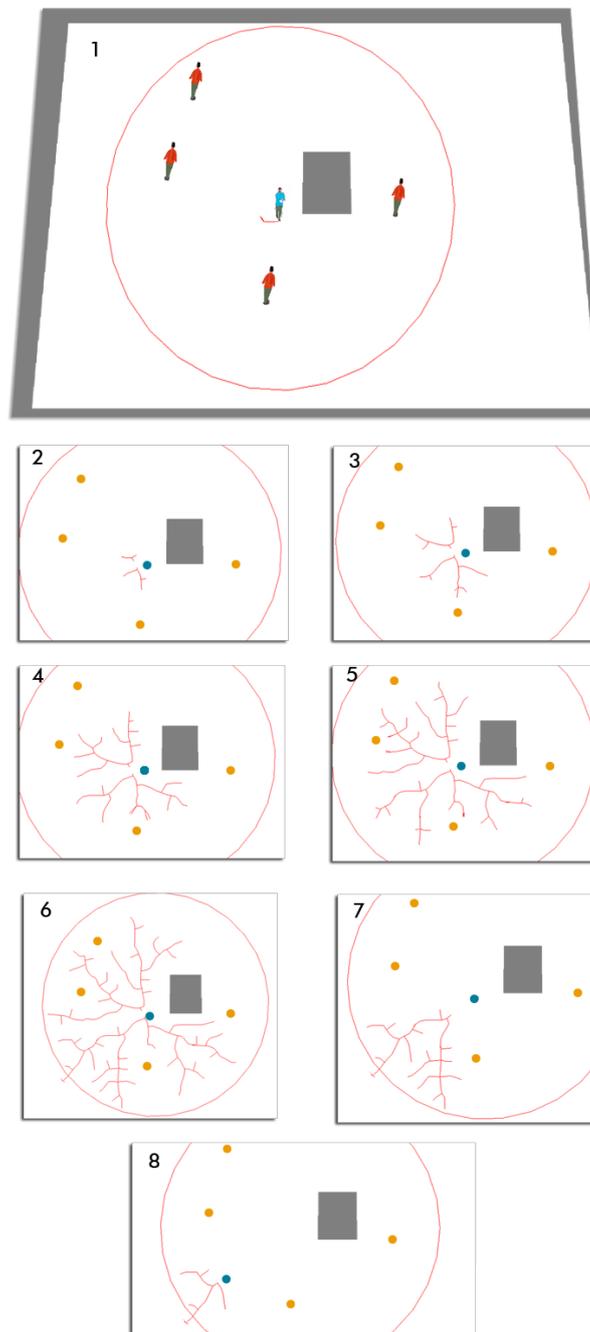


Figura 6.7: Comportamento de Fuga, agente em azul tenta escapar da área que circunda outros agentes que permanecem estáticos.

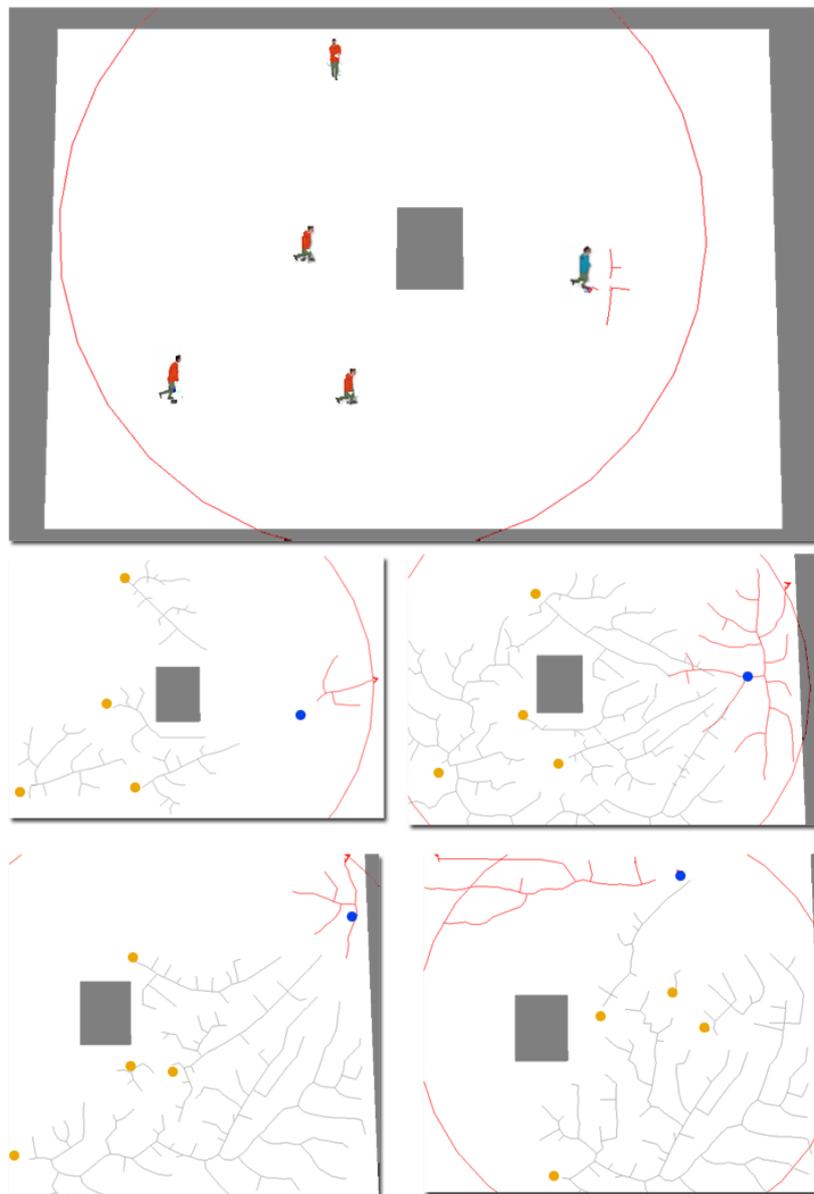


Figura 6.8: Colaboração entre agentes que perseguem um agente em fuga. Agentes à esquerda colaboram tentando alcançar o agente a direita que tenta fugir.

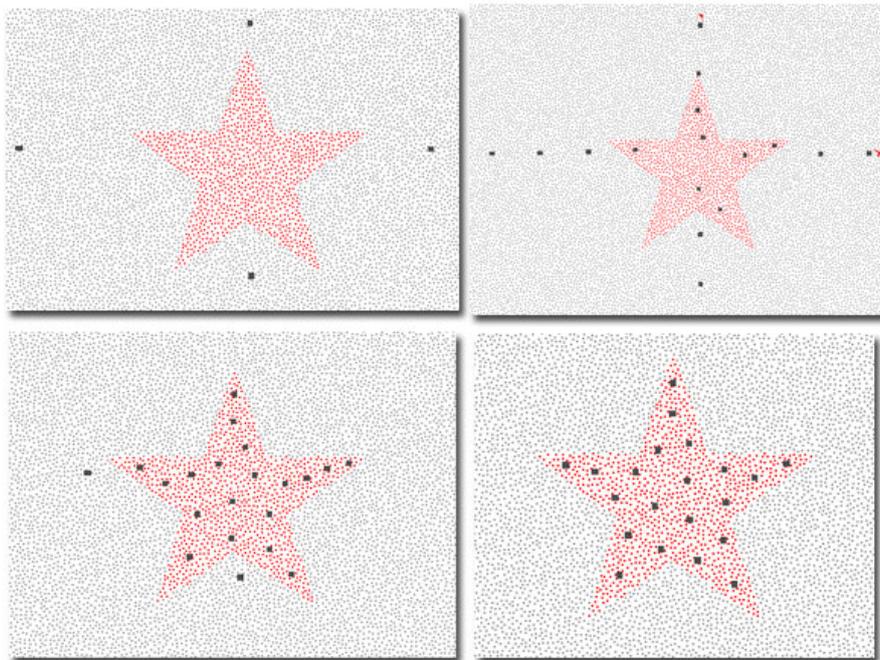


Figura 6.9: Sequência de imagens que mostram agentes chegando a uma área de formação.

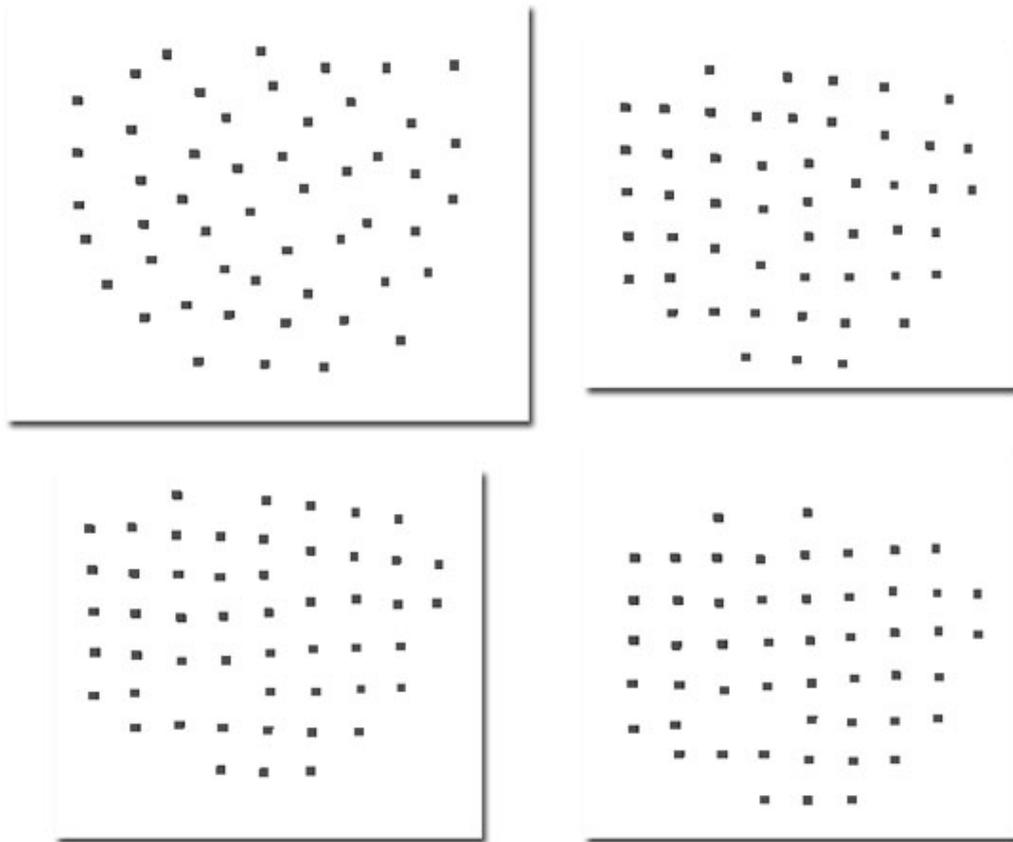


Figura 6.10: Sequência de imagens que mostram o alinhamento entre agentes.

Capítulo 7

Considerações Finais

A abordagem denominada *Tree Paths*, visou investigar se o algoritmo de canalização de nervuras que inspirou *BioCrowds*, também serviria para simular *steering behaviors* os seguintes comportamentos:

- perseguição;
- colaboração para cercar alvo;
- evasão;
- alinhamento;
- formação.

Desta forma, adaptando algumas propriedades do algoritmo de canalização de nervuras, constatou-se que é possível realizar uma rápida análise do espaço, gerando caminhos com diversidade e robustez. Assim, fugindo da homogeneidade encontrada em algoritmos de planejamento de rota convencionais que traçam sempre o melhor caminho.

Além disto, mesmo que os marcadores representem uma discretização do mundo, os caminhos gerados não necessariamente interceptam com as posições destes marcadores. Isto ocorre devido ao fato que o cálculo da posição dos nodos em *Tree Paths* serem baseados em uma resultante entre vários marcadores e seus pesos, e conforme o seu alinhamento com o objetivo. Desta forma, o agente não necessariamente se move sob marcadores, mas sim, usa estes como uma influência para traçar as rotas, possibilitando uma grande gama caminhos diferentes.

De maneira geral, *Tree Paths* mostrou-se eficiente como algoritmo de planejamento de rota para alvos fixos. Em contrapartida, observando as simulações, constata-se que a perseguição de alvos móveis não é muito eficiente, conforme as dificuldades ilustradas na Figura 4.6, e que permanecem, mesmo quando aplicada a solução da Seção 4.3.

Outras propostas para solução o problema de planejamento de rota foram levantadas, uma delas seria não mais eliminar os nodos em torno do raio de ajuste dos nodos, mas sim, escolher o nodo a ser seguido baseado em um custo que leve em conta a distância do alvo e também caminho percorrido.

Certamente, uma a solução definitiva para as rotas, seria a recomputação da árvore do agente, por inteira, a cada momento que o alvo se move. No entanto, o foco dessa dissertação é, não só estudar e adaptar algoritmo de canalização de nervuras para gerar caminhos alterando-o menos possível, como também explorar o máximo possível dos padrões gerados pelas árvores para modelar os *steering behaviors*. Portanto, optou-se por manter as ramificações das árvores o tanto quanto possível como rotas, durante a dinâmica das simulações.

O comportamento de cooperação entre agentes obteve alguns resultados visuais promissores, no entanto, conforme aumenta o número de agentes colaborando em uma determinada área, as árvores desses agentes colaborativos, que não compartilham os mesmos marcadores, podem acabar por colonizar uma quantia excessiva do espaço em busca do alvo atrapalhando a locomoção dos agentes. No ambiente de teste desta dissertação, com até 4 agentes colaborativos, obtivera-se resultados satisfatórios na área 46 x 32 m definida. Desta forma, a hipótese retirada desse experimento, é que o uso de parâmetros em *Tree Paths*, que viabilizem uma árvore com menos ramificações é uma proposta razoável para uma mais eficiente colaboração de maior número de agentes.

As simulações baseadas na adaptação do algoritmo para situações sem objetivo, onde há apenas uma região de escape (seção 4.5), obtiveram resultados promissores. *Tree Paths*, traçou bons caminhos como rotas de fuga, devido sua propriedade da colonização do espaço.

Os comportamentos emergentes de grupos, adaptados diretamente em *BioCrowds*, obtiveram resultados bastante satisfatórios, a modificação proposta para formação de agentes (ver Seção 5.1), em que os marcadores da área de formação passam a ter o mesmo peso, demonstrou a coesão de *BioCrowds* quanto a sua aplicação da teoria da competição pelo espaço, pois, conforme pode ser observado na Figura 6.9, esta se encarrega de distribuir de maneira eficiente os agentes pela área. Já o comportamento de alinhamento (ver Seção 5.2) pode apresentar alguns erros, caso os agentes envolvidos estejam muito próximos no início do processo.

Existe uma série de dificuldades em avaliar de forma sistemática esta categoria de comportamentos denominados *steering behaviors*. Isto se dá devido a complexidade destes comportamentos, através de critérios como o tempo, distância percorrida, linearidade que podem não ser necessariamente parâmetros para decidir se um modelo é bom ou não.

No contexto dessa dissertação, o algoritmo *Tree Paths*, mostrou-se pouco sugestível a criar caminhos lineares e otimizados para todas as situações, desta forma, uma comparação com outros algoritmos semelhantes como o [LAV00] ainda é necessária como trabalhos futuros. *Tree Paths* é apto a oferecer comportamentos diversificados para aplicações em grupos de agentes a cada simulação, ainda com coerência ao contexto. Esta característica é alcançada, não devido a aleatoriedade de algumas equações, mas sim, por suas regras que fazem com que o conjunto de caminhos tenha sempre uma configuração diferente, mesmo quando aplicado ao mesmo ambiente em mesma situações.

Esta característica também é encontrada no modelo original [RUN05], capaz de gerar diferentes canalizações de nervuras para plantas ou superfície de folhas, conforme pode ser observado na Figura 2.10.

Referências Bibliográficas

- [BOV96] Bouvier, E.; Guilloteau, P. "Crowd simulation in immersive space management". London, In: Proceedings of the Eurographics workshop on Virtual environments and scientific visualization '96, 2006, pp. 104-110.
- [COO86] Hall, E. T. "Stochastic sampling in computer graphics". ACM Transaction. Graphics, Vol 5, issue 1, 1986, pp. 51-72.
- [DON00] Tomaz, G.; Donikian, S. "Modeling virtual cities dedicated to behavioural animation". The International Journal of The Eurographics Association (Computer Graphics Forum)", 2000, pp. 71-80.
- [FOW92] Fowler, D. R.; Meinhardt, H.; Prusinkiewicz, P. "Modeling seashells". In: Proceedings Computer Graphics SIGGRAPH, 1992, pp. 379-387.
- [GOL01] Goldenstein, S. et al. "Scalable nonlinear dynamical systems for agent steering and crowd simulation". Computers and Graphics, Vol 23, issue 6, 2001, pp. 983-998.
- [HAL59] Hall, E. T. "The Silent Language". Anchor, New York, 1973, 240p.
- [HEL00] Helbing, D.; Farkas I.; Vicsek, T. "Simulating dynamical features of escape panic". Nature, Vol 407, 2000, pp. 487-490.
- [HEL03] Werner T.; Helbing D. "The social force pedestrian model applied to real life scenarios". Greenwich, Pedestrian and Evacuation Dynamics 2003, pp. 17-26.
- [ISC65] Rufus, I. "Differential Games: A Mathematical Theory with Application to Warfare and Pursuit, Control and Optimization". John Wiley and Sons, New York, 1965, 416p.
- [LAV00] LaValle, S. et al, "Rapidly-exploring random trees: Progress and prospects". In: Proceedings Workshop on the Algorithmic Foundations of Robotics, 2000, p. 293-308.
- [LIN07] Avneesh, S.; Gayle, R.; Andersen, E.; Guy, S.; Lin, M.; Manocha, D. "Real-time navigation of independent agents using adaptive roadmaps". In: VRST '07: Proceedings of the 2007 ACM symposium on Virtual reality software and technology, 2007, pp. 99-106.

- [LOG98] Logan, B.; Alechina, N. "A* with bounded costs". AAAI Press/MIT Press, Birmingham , 1998, 6p.
- [MUS01] Thalmann, D.; Musse, S. "Hierarchical model for real time simulation of virtual human crowds". IEEE Transactions on Visualization and Computer Graphics, Vol 7, issue 2, 2001, pp. 152-164.
- [MUS09] Musse, S.; Jung, C.; Bicho A.; Paravisi M.; Rodrigues, R. "Steering Behaviors Based on Colonization Algorithm". In: Proceedings of Computer Graphics International, 1998.
- [MUS97] Musse, S. R.; Thalmann D. "A Model of Human Crowd Behavior: Group Inter-Relationship and Collision Detection Analysis". In: Workshop Computer Animation and Simulation of Eurographics, 1997, pp. 39-52.
- [NOR48] Wiener, N. "Cybernetics: or control and Communication in the Animal and the Machine". The M.I.T. Press, New York, 1948, 318p.
- [OKA08] Okabe, A.; Boots, B.; Sugihara, K.; Chi, S. N. "Spatial Tessellations: Concepts and Applications of Voronoi Diagrams". Wiley, 2000, West Sussex, 696p.
- [PRU94] Prusinkiewicz, P. "Visual models of morphogenesis". In: Artificial Life, Vol. 1, Issue 1 1994, pp. 67-74.
- [REE83] Reeves, W. T. "Particle Systems - A Technique for Modeling a Class of Fuzzy Objects". CM Transactions Graphics, 1983, pp. 91-108.
- [REY87] Reynolds, C. "Flocks, herds and schools: A distributed behavioral model". In SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques. pp. 25-34.
- [REY99] Reynolds, C. "Steering Behaviors for Autonomous Characters". In: Game Developers Conference, 1999.
- [ROD09] Rodrigues, R.; Bicho, A.; Paravisi, M.; Musse, S.; Jung, C.; Magalhaes, L. "Tree Paths: A New Model for Steering Behaviors". In: Proceedings IVA'09, 2009, pp. 358-371.
- [RUN05] Runions, A.; Fuhrer, M.; Lane, B.; Federl, P.; Rolland-Lagan, A.; Prusinkiewicz, P. "Modeling and visualization of leaf venation patterns". In: Proceedings of SIGGRAPH, 2005, 10p.
- [STE94] Stewart, W. J. "Introduction to the numerical solution of Markov chains". Princeton University Press, 1994, 539p.

- [STI00] Still, K. G. "Crowd Dynamics". Ph. D. Thesis, University of Warwick, Coventry, UK, 2000, 280p.
- [STI94] Still, G. K. "New Insights into Crowd Behaviour - It's Fractal."Focus, nov/94, 30p.
- [TEU06] Treuille, A.; Cooper, S.; Popovic, Z. "Continuum crowds". ACM Transactions on Graphics, Vol 25, 2005, pp. 1160-1168.
- [TUR91] Turk, G. "Generating textures on arbitrary surfaces using reactiondiffusion". In: Proceedings Computer Graphics SIGGRAPH, 1991, pp. 289-298.