

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Modelo para Simulação de Multidões Utilizando Lógica Nebulosa

Tharak Zuchetto Nasser

Florianópolis - SC

2011/2

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

Modelo para Simulação de Multidões Utilizando Lógica Nebulosa

Tharak Zuchetto Nasser

Trabalho de conclusão de curso
apresentado como parte dos
requisitos para obtenção do grau de
Bacharel em Ciências da Computação.

Florianópolis - SC

2011/2

Tharak Zuchetto Nasser

Modelo para Simulação de Multidões Utilizando Lógica Nebulosa

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Ciências da Computação.

Orientador: Dr. Mauro Roisenberg

Banca examinadora:

Ricardo Azambuja Silveira

João Candido Lima Dovicchi

Sumário

Lista de Figuras.....	6
Resumo.....	7
Abstract.....	8
1. Introdução.....	9
1.1 Motivação e justificativa.....	9
1.2 Objetivos.....	10
1.3 Objetivos específicos.....	10
1.4 Metodologia.....	11
2. Fundamentação Teórica.....	12
2.1 Modelo de Helbing.....	12
2.2 Modelo de percepção por visão para um agente.....	13
2.3 Detecção e desvio de obstáculos e agentes.....	15
2.4 Flocking.....	15
2.5 Lógica Nebulosa.....	18
2.5.1 - Avaliar o antecedente de cada regra.....	21
2.5.2 - Obter a conclusão de cada regra.....	23
2.5.3 - Agregar as conclusões.....	24
2.5.4 - Defuzzificação.....	25
2.5.5 - Resumo e conclusão.....	26

2.6 NetLogo.....	27
3. Desenvolvimento.....	29
3.1. Modelo Proposto.....	29
4. Validação.....	32
4.1 Resultados Obtidos.....	32
4.1.1 Corredor com Alargamento.....	32
4.1.2 Cruzamento.....	34
4.1.3 Centro da Cidade.....	35
4.1.4 Ambiente sem Obstáculos.....	36
5. Conclusão.....	40
5.1 Trabalhos Futuros.....	40

Lista de Figuras

Figura 1 - Imagem retirada de [7], fórmula utilizada no modelo de Helbing.....	14
Figura 2 - Imagem retirada de [7], que mostra o resultado gerado utilizando o modelo de Helbing.....	15
Figura 3 - Imagem retirada de [2], onde Um agente percebe outro dentro de sua área de percepção.	16
Figura 4 - Imagem retirada de [10], detecção de colisão utilizando o modelo proposto por [10].....	17
Figura 5 - Imagem retirada de [17]. Etapa <i>input fuzzyfication</i>	23
Figura 6 - Imagem retirada de [17]. Obter a conclusão de cada regra.....	24
Figura 7 - Imagem retirada de [17]. Agregando conclusões.....	25
Figura 8 - Imagem retirada de [17]. Defuzzificação.....	26
Figura 9 - Imagem retirada de [17]. Todas as etapas do método de Mamdani.....	27
Figura 10 - Cone com raio 5 e ângulo 60.....	31
Figura 11 – Conjuntos utilizados na Lógica Nebulosa.....	31
Figura 12 – Regras de inferência do modelo proposto.....	32
Figura 12 – Afastar-se do grupo. Adaptada de [20].....	35
Figura 13 – Alinhar-se em relação ao grupo. Adaptada de [20].....	35
Figura 14 – Mover-se em direção ao centro do grupo. Adaptada de [20].....	36
Figura 15: Conjunto nebuloso genérico.....	36

Figura 16 - Imagem retirada de [19] que mostra o resultado de uma multidão passando por um corredor que possui uma área onde ele se alarga.....	38
Figura 17 - Resultado obtido no modelo proposto.....	38
Figura 18 - Imagem retirada de [18] que mostra o resultado da simulação em um cruzamento.....	39
Figura 19 - Resultado da simulação utilizando o modelo proposto.....	40
Figura 20 - Imagem retirada de [18] que mostra o resultado da simulação no centro da cidade.....	40
Figura 21 - Resultado da simulação utilizando o modelo proposto.....	41
Figura 22 - Estados da simulação sem obstáculos.....	43
Figura 23 - Formação de um grande grupo após algum tempo de simulação.....	44

Resumo

As simulações do comportamento humano em situações de pânico são utilizadas no planejamento arquitetônico de ambientes onde existe um grande fluxo de pessoas e, a partir dos resultados gerados por elas, é possível prever, com certo grau de realismo, como uma multidão se comportaria no caso de haver necessidade de evacuar o local e possibilitando que seja feita uma melhoria no local para facilitar a evacuação do mesmo.

Esse trabalho propõe desenvolver um modelo de simulação que imite o comportamento de pessoas de uma forma mais realista através da utilização de Lógica Nebulosa no lugar das fórmulas matemáticas que definem o comportamento dos agentes que representam seres humanos.

Palavras-chave: simulação de comportamento humano, situações de pânico, simulação, lógica nebulosa.

Abstract

Simulations of human behavior in panic situations are used in architectural planning environments where there is a large flow of people and from the results generated by these can foresee with some degree of realism, as a crowd would behave in the event of an emergency , allowing the location to be improved to facilitate the evacuation of it.

This work proposes to develop a simulation model that mimics the behavior of people in a more realistic way with the use of fuzzy logic instead of the mathematical formulas that define the behavior of the agents that represent humans.

Keywords: simulation of human behavior, panic situations, simulation, fuzzy logic.

1. Introdução

Em simulações de comportamentos de multidões são utilizadas fórmulas complexas na implementação do comportamento dos agentes. E essas implementações são baseadas em fenômenos físicos, como soma dos vetores das forças que estão sendo aplicadas em um agente. A ideia desse trabalho é que se utilize lógica nebulosa para substituir essas regras, uma vez que acredita-se que, com lógica nebulosa é mais fácil de simular o comportamento de um ser humano.

Na lógica nebulosa não se usam valores fixos para representar um valor. É criado um conjunto para representá-lo, por exemplo, quando se pergunta qual a distância entre dois pontos sem a utilização de lógica nebulosa a resposta será em metros, já com lógica nebulosa será distância curta, distância média ou distância longa.

Seres humanos utilizam uma linguagem natural e percepções mentais para decidir que ações vão tomar, chegando a conclusões expressas em palavras. Estas não possuem um significado exato, assim como as palavras usadas na representação de um sistema nebuloso. Portanto, acredita-se que, utilizando lógica nebulosa, será possível obter um resultado mais próximo ao real.

1.1 Motivação e justificativa

Em simulações de comportamento humano são utilizadas fórmulas matemáticas

para calcular como uma entidade, que representa um ser humano, vai reagir levando em consideração o estado atual do ambiente em que ela se encontra.

Com a utilização de lógica nebulosa espera-se acrescentar mais realismo à simulação que vai ajudar a melhorar o planejamento urbanístico arquitetônico de um determinado local, tornando-o mais seguro, pois será possível perceber eventuais obstáculos que possam dificultar a evacuação deste local caso ocorra alguma situação de emergência.

1.2 Objetivos

Desenvolver um modelo de evacuação de multidões onde o agente representa uma pessoa que toma decisões, baseadas na sua situação atual, de forma mais natural do que nos modelos já existentes.

Esse trabalho pretende mostrar como a utilização de lógica nebulosa pode melhor aproximar, em relação à maneira convencional, o comportamento de uma entidade programada ao de um ser humano. Para isso será desenvolvido um programa que utilize lógica nebulosa na implementação do comportamento de seus agentes.

1.3 Objetivos específicos

Realizar um pesquisa sobre os atuais modelos de simulação de comportamento de multidões.

Implementar um modelo de comportamento de multidões utilizando lógica nebulosa.

Verificar se a implementação que utiliza lógica nebulosa fornece um resultado aceitável em relação aos modelos já conhecidos na área de simulação de comportamento de agentes em multidões.

1.4 Metodologia

Este trabalho está dividido em quatro etapas.

- Etapa 1 - Fundamentação Teórica: Foi feito um estudo sobre modelos que são utilizados em simulações de comportamento de pessoas em multidões, e também de modelos para simular o comportamento dessas multidões em situações de pânico. Também foi feita uma pesquisa sobre lógica nebulosa e suas aplicações, ao final dessa pesquisa é explicado o método que melhor se aplicou ao modelo proposto neste trabalho. No final dessa etapa é falado sobre o programa que foi utilizado para implementar o modelo proposto.
- Etapa 2 - Desenvolvimento: Nesta etapa é apresentado o modelo proposto que foi criado a partir das pesquisas feitas na etapa anterior. Em seguida é explicado como ocorreram as simulações, assim como seus

parâmetros. Ao final são mostrados os cenários que foram utilizados para testar a eficiência do modelo proposto.

- Etapa 3 - Validação: Primeiramente são definidos os parâmetros da avaliação que decidirá se o modelo proposto atinge seus objetivos. Depois são simulados os cenários definidos anteriormente utilizando-se o modelo proposto. Após a coleta de dados é feita a avaliação.
- Etapa 4 - Conclusão: Aqui são mostrados as conclusões geradas a partir dos resultados obtidos nas simulações feitas na etapa anterior.

1.5 Estrutura dos Capítulos

Capítulo 2 é feita a fundamentações teórica onde são mostrados os modelos que foram utilizados como base para a implementação do modelo deste trabalho.

Capítulo 3 descreve o modelo proposto e seu funcionamento.

Capítulo 4 descreve os experimentos que foram realizados para que fosse possível a validação do modelo.

Capítulo 5 apresenta a conclusão chegada a partir dos experimentos e sugere melhorias que podem ser feitas para melhorar o modelo.

2. Fundamentação Teórica

Neste capítulo serão apresentados os modelos e regras que influenciaram diretamente no desenvolvimento deste trabalho. Primeiramente é mostrado o modelo que serviu como início para este trabalho, o modelo de Helbing. Tendo esse modelo como base foram agregadas características de outros modelos, como área de percepção e detecção e desvio de obstáculos. Foi adicionado também o comportamento de *flocking*, para que os agentes se comportassem de forma mais natural. Após isso os cálculos que existiam no modelo foram substituídos pela Lógica Nebulosa.

2.1 Modelo de Helbing

Helbing [7] propôs um modelo baseado em física e forças sócio-psicológicas para descrever o comportamento de multidões em situações de emergência. Ele utiliza um sistema onde cada partícula i de massa m_i possui uma velocidade v , ou seja, a velocidade com a qual se deslocará em direção ao seu objetivo corresponde ao vetor e_i . Essa velocidade é recalculada a cada intervalo de tempo t_i . Simultaneamente as partículas tendem a manter uma distância de outras entidades, j , e de paredes, w , seus vetores direção são f_{ij} e f_{iw} , respectivamente. O cálculo da nova velocidade no tempo t se dá pela equação da imagem a seguir.[1]

$$m_i \frac{d\vec{v}_i}{dt} = m_i \underbrace{\frac{v_i^o(t)\vec{e}_i^o(t) - \vec{v}_i(t)}{\tau_i}}_{1^{st}} + \underbrace{\sum_{j \neq i} \vec{f}_{ij}}_{2^{nd}} + \underbrace{\sum \vec{f}_{iw}}_{3^{rd}}$$

Figura 1 - Imagem retirada de [1], fórmula utilizada no modelo de Helbing [7]

Esse modelo gera fenômenos realistas como agentes se aglomerando (Figura 2) ao redor de uma saída e o aumento do tempo de evacuação do local quando a velocidade desejada é aumentada, como afirma Helbing.

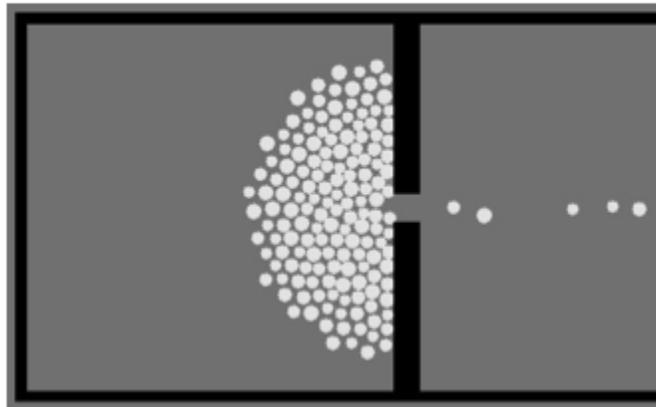


Figura 2 - Imagem retirada de [7], que mostra o resultado gerado utilizando o modelo de Helbing [7]

2.2 Modelo de percepção por visão para um agente

O modelo construído em A Perceptive Model for Virtual Agents in Crowds utiliza uma abordagem de zonas para simular a percepção visual do agente. Essa região possui dois atributos, distância, d , e ângulo, θ , que são fixados no início da simulação. Outro agente ou um obstáculo será percebido quando entrar nessa região de percepção do agente (Figura 3). [2]

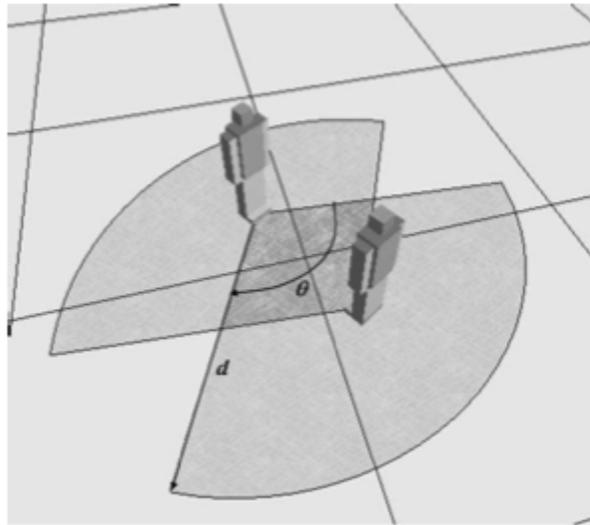


Figura 3 - Imagem retirada de [2], onde um agente percebe outro dentro de sua área de percepção.

2.3 Detecção e desvio de obstáculos e agentes

O sistema proposto por SHARMA [10][11][12] utiliza um método de detecção de colisão com base na distância. Isso envolve o uso de distância entre dois objetos para determinar se os agentes colidiram. Este método favorece círculos perfeitos, mas pode ser melhorado para outras formas também.

A colisão é verificada pelo agente que está se movendo em direção ao objetivo.

Quando a distância entre o agente e um outro agente fica abaixo de um valor limite, o método responsável por calcular o desvio para não haver uma colisão entre os agentes é chamado. Em [2] é utilizada uma abordagem similar, como o agente fica responsável por verificar eventuais colisões esta é tratada sem muito custo de processamento, diferente do que aconteceria se existisse uma entidade global verificando a cada momento se algum agente está em rota de colisão com outro. Como resultado é gerada uma animação mais suave e natural.

Em [10] obstáculos são representados por círculos. Cada círculo possui uma área ao seu redor e esta é dividida em 12 partes iguais, quando um agente entra nessa área ele muda sua direção para a próxima área mais perto e assim por diante até que o obstáculo seja contornado (Figura 4). Após ter desviado o ângulo do agente é recalculado para apontar para o objetivo.

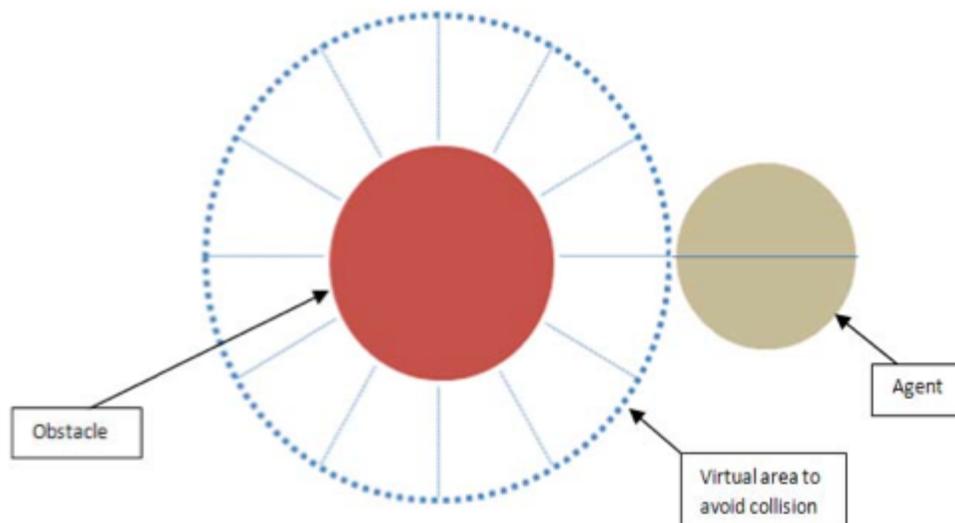


Figura 4 - Imagem retirada de [10], detecção de colisão utilizando o modelo proposto por [10]

2.4 Flocking

A movimentação de bandos de pássaros e grupos com comportamento relacionados, como cardumes de peixe ou manadas de animais, é tão sincronizada que parece que existe uma força de controle sobre o grupo, mas na verdade isso é causado devido ao comportamento individual de cada membro do grupo, cada um agindo de acordo com sua percepção local do mundo.

Uma das áreas de interesse em animação gráfica é a descrição e controle desse tipo de comportamento. Programadores da área buscam inventar novos tipos de abstração de movimentos e duplicar o movimento do mundo real. A princípio criar uma simulação de um bando apresenta serias dificuldades. Programar o caminho de um grande número de objetos individuais usando técnicas tradicionais de animação seria muito tedioso. Dado a complexidade do caminho que cada membro do grupo deve percorrer, é pouco provável que não existiriam erros na implementação do movimento, e também seria muito difícil editar os parâmetros da animação. [15]

Dados esses problemas, Reynolds[15] sugeriu uma abordagem eficiente, robusta e condizente com a realidade que se baseia na ideia de que o movimento de um grupo, na verdade, é gerado pelo comportamento individual de cada membro do mesmo. Nessa abordagem, simulou-se uma parte do comportamento de um pássaro responsável pelo comportamento que permite que ele faça parte de um bando. Além da parte responsável pela percepção e aspectos físicos do vôo.

Em [15] foi usada uma abordagem baseada em sistemas de partículas, onde partícula correspondia a um pássaro. Cada pássaro tentava, independentemente e ao mesmo tempo, ficar junto de seu bando e evitar colisões com seus companheiros e objetos do ambiente.

Para conseguir simular o comportamento de grupos, Reynolds dividiu o comportamento dos agentes em três etapas: evitar colisão, igualar velocidade e ir para o centro do grupo.

No algoritmo para evitar colisão leva-se em consideração a posição dos membros que estão a certa distância, e para igualar a velocidade, a velocidade dos mesmos.

Como cada agente tem uma percepção diferente o centro do grupo, na verdade, é usado o centro dos agentes que estão dentro da área de percepção dele para calcular a direção que ele deve tomar para este centro local. Grupos na vida real se separam quando tentam se desviar de um obstáculo, membros do grupo não se preocupam se nem todos os membros estão indo na mesma direção, somente os que estão dentro de sua área de percepção.

Esses comportamentos que compõem o modelo estão expressos em questão de "membros próximos", que são os agentes que estão em uma área circular centralizada no agente.

A parte mais interessante da simulação de um grupo, segundo Reynolds, é a interação com outros objetos do ambiente. O comportamento isolado do grupo tende

a alcançar uma estabilidade. Por exemplo, os conflitos entre evitar colisão e ir para o centro do grupo não levam a um conflito, mas sim a um equilíbrio. Obstáculos no ambiente e a tentativa do grupo de desviar deles aumentam a aparente complexidade do comportamento do grupo.

Ainda segundo Reynolds, somar um valor ao ângulo do vetor direção do agente é o melhor jeito de simular um agente guiado pela visão pois ele considera somente os obstáculos presentes na sua área de percepção.

2.5 Lógica Nebulosa

“A teoria dos conjuntos nebulosos foi desenvolvida a partir de 1965 por Lotfi Zadeh[6], para tratar do aspecto vago da informação.” [16] Quando aplicada em sistemas baseados em conhecimento é conhecida como lógica nebulosa, lógica difusa ou lógica fuzzy.

De acordo com [6], a principal contribuição da lógica nebulosa é poder programar utilizando palavras. Nenhuma outra metodologia, como, lógica de predicados, teoria probabilística, teoria de redes neurais, redes Bayesianas, etc., servem para esse propósito. Zadeh chama isso de *Computing with Words* (CW). Claro que essa não é a única utilidade de lógica nebulosa, mas é a que melhor se aplica no contexto deste trabalho.

Para explicar como funciona a lógica nebulosa é preciso que fiquem claras algumas definições. Primeiramente o conceito de granularidade. Grânulo, basicamente,

é um conjunto nebuloso de pontos que possuem a forma de um aglomerado de elementos unidos por alguma similaridade. Uma palavra p é um rótulo de um grânulo g e, reciprocamente, g é a denotação de p . Uma palavra pode ser simples (como em jovem) ou composta (como em não muito jovem). [6]

O papel da lógica nebulosa em CW é propagar as premissas para conclusões. Dada uma entrada x , verifica-se em que conjunto nebuloso ele se encontra. Por exemplo, muito longe, através da lógica nebulosa, será transformado em uma saída, aumentar velocidade.

Considere a preposição Maria é jovem. Nesse caso jovem é o rótulo do grânulo jovem (note que, por simplicidade, o mesmo símbolo é usado para ambos, a palavra e sua denotação). O conjunto nebuloso jovem tem o papel de restrição da idade de Maria. [6]

Para cada grânulo existem graus de compatibilidade, esse grau varia de 0 à 1, sendo que com grau igual a 1 o objeto é completamente compatível com o conjunto, com grau maior que 0 e menor que 1 o objeto é parcialmente compatível e com grau igual a 0, completamente incompatível. Para pertencer ao grânulo jovem do exemplo anterior a pessoa deve ter idade entre 15 e 25 anos, se Maria tiver 14 anos ela será incompatível com jovem, mas poderá ser compatível com outro grânulo. Se ela tem 16 anos será compatível parcialmente, dependendo da função que define o grânulo. E será completamente compatível se tiver idade igual a 20 anos.

As noções de inclusão, união, intersecção, complemento, etc, são extendidas

para conjuntos nebulosos e várias propriedades dessas noções são aplicáveis neles.

[6]

Através de um modelo matemático e um conjunto de regras, os valores de entrada são transformados em graus de compatibilidade em relação aos grânulos, para serem usados em uma base de conhecimento.

“A base de conhecimento consiste de uma base de dados e uma base de regras, de maneira a caracterizar a estratégia de controle e as suas metas.” [16]

A base de dados possui os graus que foram gerados pelo modelo matemático. Já a base de regras é formada por estruturas do tipo, se <premissa> então <conclusão>, como, por exemplo: Se Maria é jovem então ela é solteira. No caso de existirem várias premissas elas serão relacionadas utilizando conjunção (e) ou disjunção (ou), gerando um grau de compatibilidade global.

Após geradas todas as regras, existirão algumas conclusões, e estas serão tratadas para que o sistema possa criar uma única saída. Para isso, existem três métodos clássicos. Primeiro Máximo, onde encontra-se o valor de saída através do ponto em que o grau de pertinência da distribuição da ação de controle atinge o primeiro valor máximo. Método da Média dos Máximos, onde calcula-se o ponto médio entre os valores que têm o maior grau de pertinência inferido pelas regras. E o Método do Centro da Área, que terá como saída o centro de gravidade da função de distribuição de possibilidade da ação de controle.

Um dos principais métodos utilizados na implementação da lógica nebulosa

é o método de Mamdani, devido a simplicidade de suas operações que utilizam, basicamente, mínimos e máximos.

Este método pode ser dividido em quatro etapas que serão explicadas a seguir de acordo com o exemplo retirado de [17].

2.5.1 - Avaliar o antecedente de cada regra

Dados os valores de entrada obtém-se valores associados aos conjuntos fuzzy. Esse processo é chamado de *input fuzzyfication*. Se o antecedente da regra tem mais de uma parte, um operador nebuloso (*AND* ou *OR*) é aplicado para obter um único valor de associação.

Dados os antecedentes *service* com valor 3 e *food* com valor 8, quando se fuzzifica *service* em relação ao conjunto nebuloso *service is excelente* obtém-se um valor de associação 0. Já o outro antecedente, *food*, com o conjunto *food is delicious* gera um valor de associação 0.7.

Finalmente as duas partes do antecedente são unidas através de uma disjunção (*service is excelente* OU *food is delicious*), aplica-se a operação OU, o máximo, nos dois valores de associação para obter um novo valor de associação, 0,7. Supondo que os dois valores do antecedente fossem unidos pelo conjunção E utilizaria-se a operação E, tendo como resultado 0.

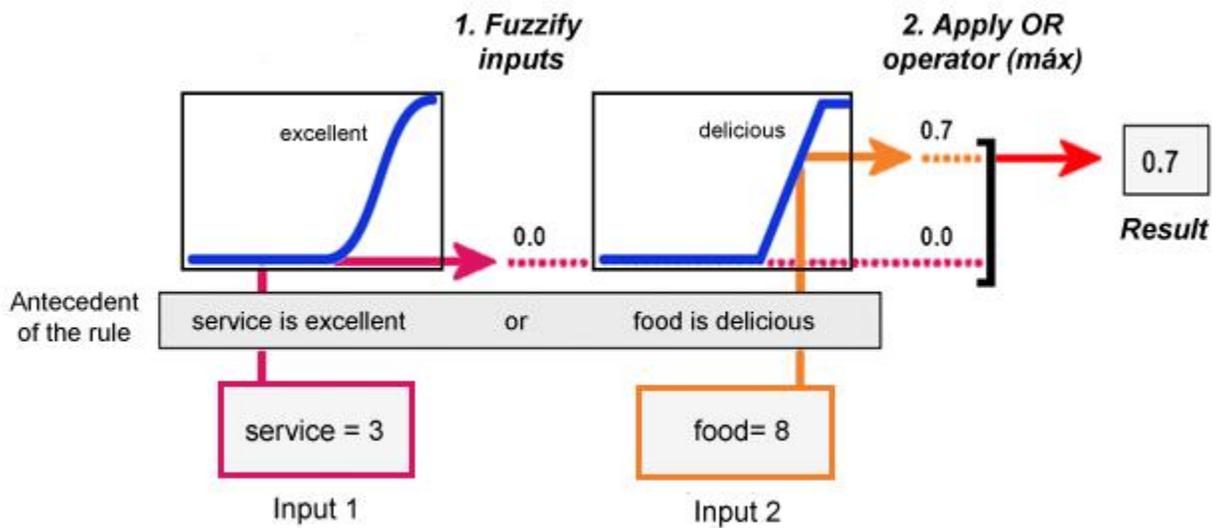


Figura 5 - Imagem retirada de [17]. Etapa *input fuzzyfication*

2.5.2 - Obter a conclusão de cada regra

Dado o consequente de cada regra (um conjunto fuzzy) e o valor do antecedente obtido na etapa anterior, aplica-se um operador de implicação nebuloso para obter um novo conjunto fuzzy.

Os dois métodos mais usados para isso são o mínimo, que trunca o consequente da função de pertinência, e o produto, que a dimensiona. No exemplo abaixo é utilizado o operador máximo.

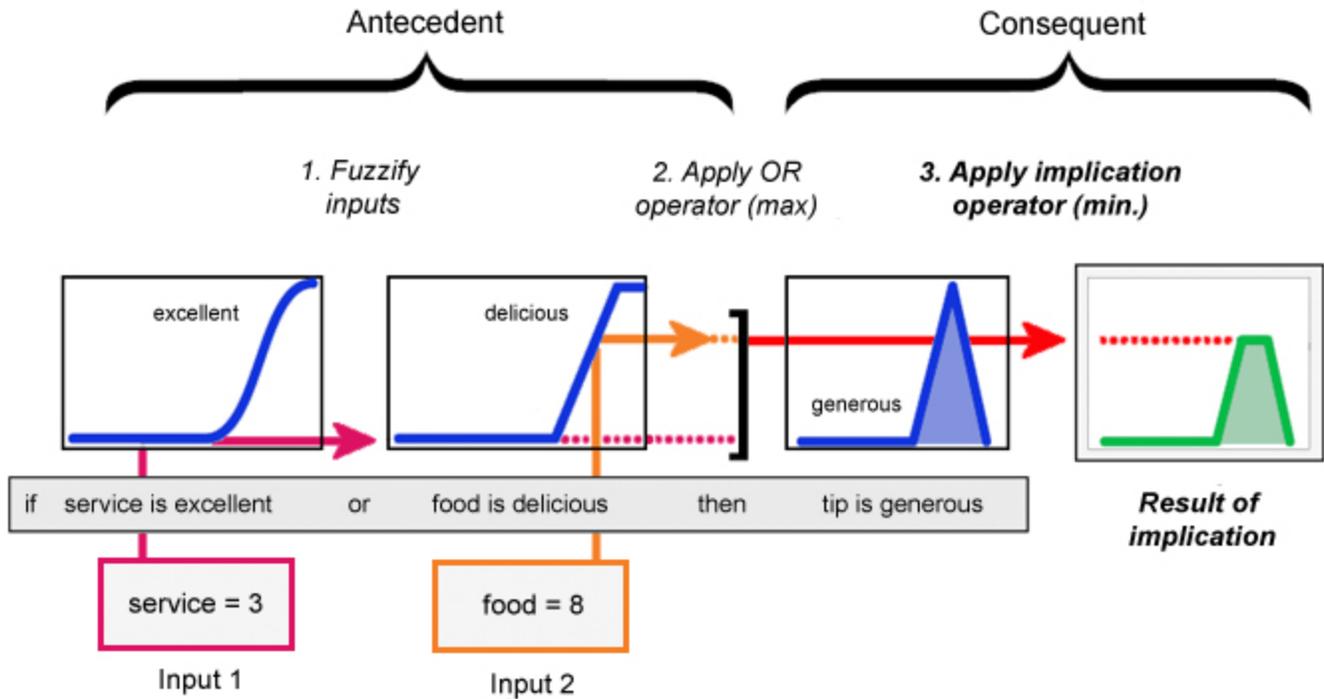


Figura 6 - Imagem retirada de [17]. Obter a conclusão de cada regra

SE *service is excellent* (0) OU *food is delicious* (0.7) ENTAO *tip is generous* (0.7).

2.5.3 - Agregar as conclusões

Nessa etapa, combina-se as saídas geradas anteriormente em um único conjunto fuzzy, utilizando-se um operador de agregação fuzzy.

Os operadores mais utilizados são: o operador de máximo, o operador de soma e o de soma probabilíssima. No exemplo abaixo utiliza-se o operador de máximo.

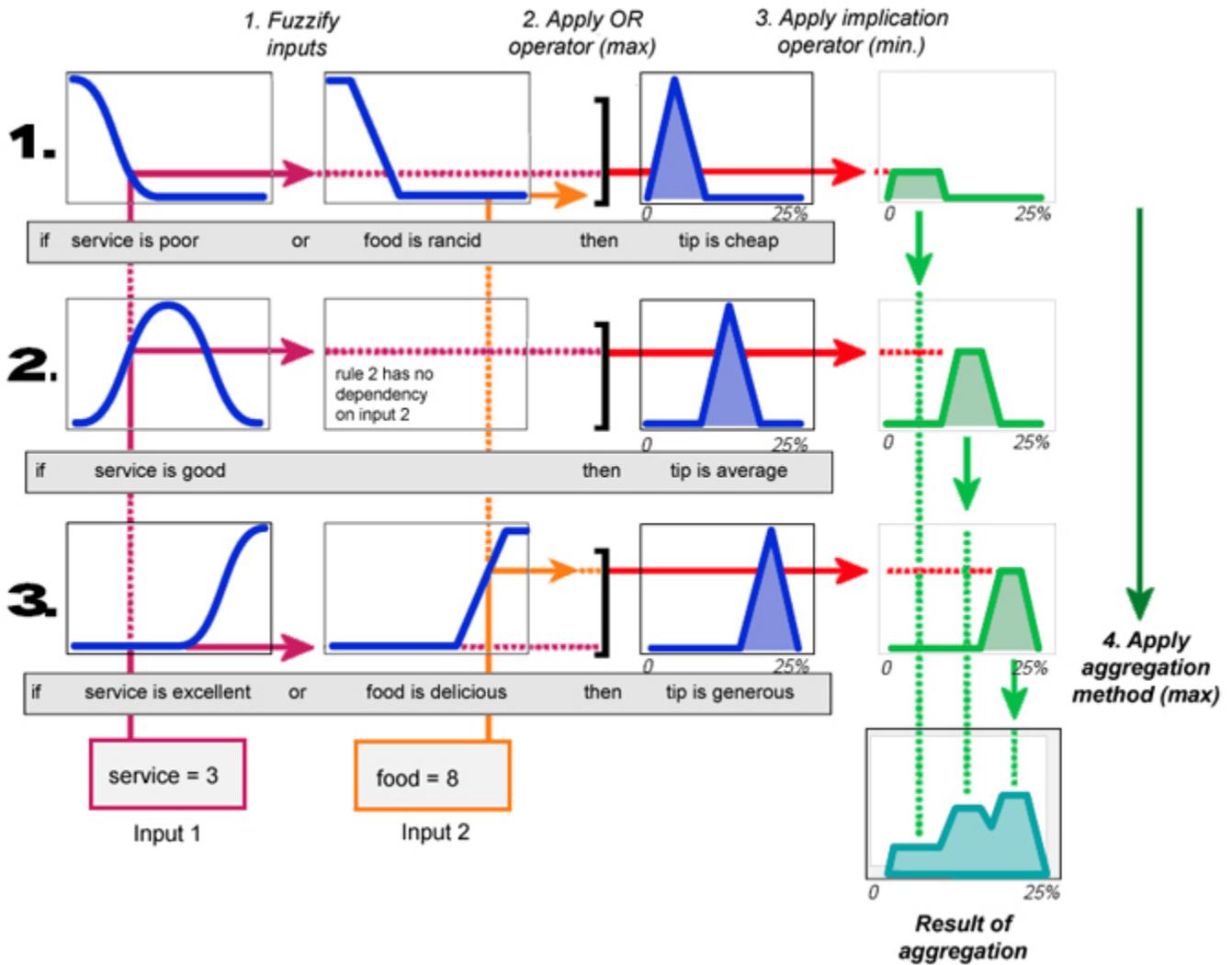


Figura 7 - Imagem retirada de [17]. Agregando conclusões

2.5.4 - Defuzzificação

Quando tentamos resolver um sistemas de decisão, queremos que a saída seja um número e não um conjunto fuzzy. Seguindo o exemplo mostrado, não queremos que o sistema nos diga para dar uma gorjeta generosa, queremos que nos de o valor

da gorjeta. Então é necessário transformar o conjunto nebuloso que foi obtido na etapa anterior em um único valor numérico. Um dos métodos de defuzzificação mais popular é o centróide, que retorna ao centro da área sob o conjunto nebuloso obtido anteriormente. Os cálculos são mostrados na Figura 8:

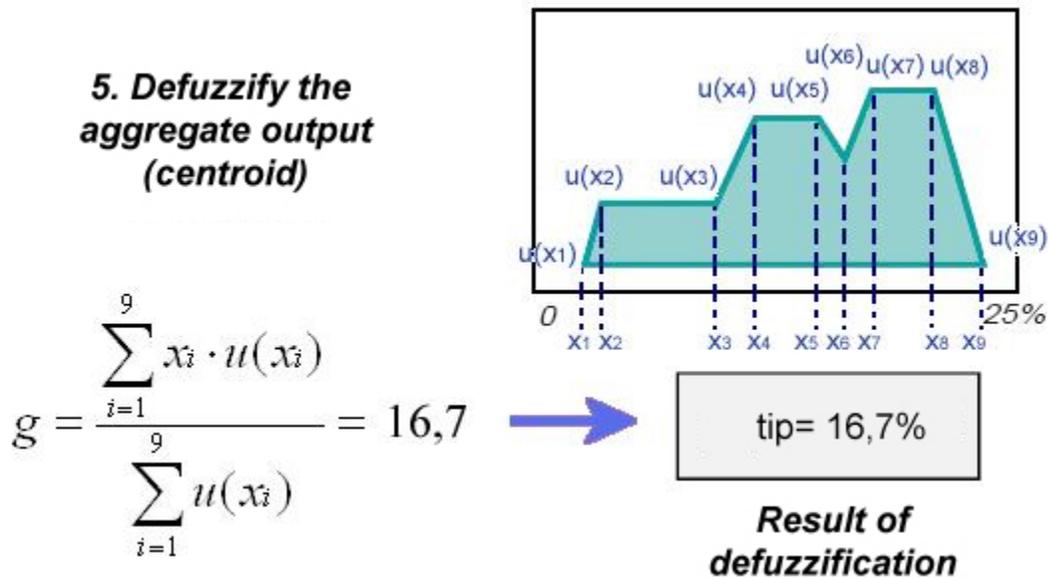


Figura 8 - Imagem retirada de [17]. Defuzzificação

2.5.5 - Resumo e conclusão

Na Figura 9 é possível ver todas as etapas do processo de fuzzyficação e defuzzyficação utilizando o método de Mamdani:

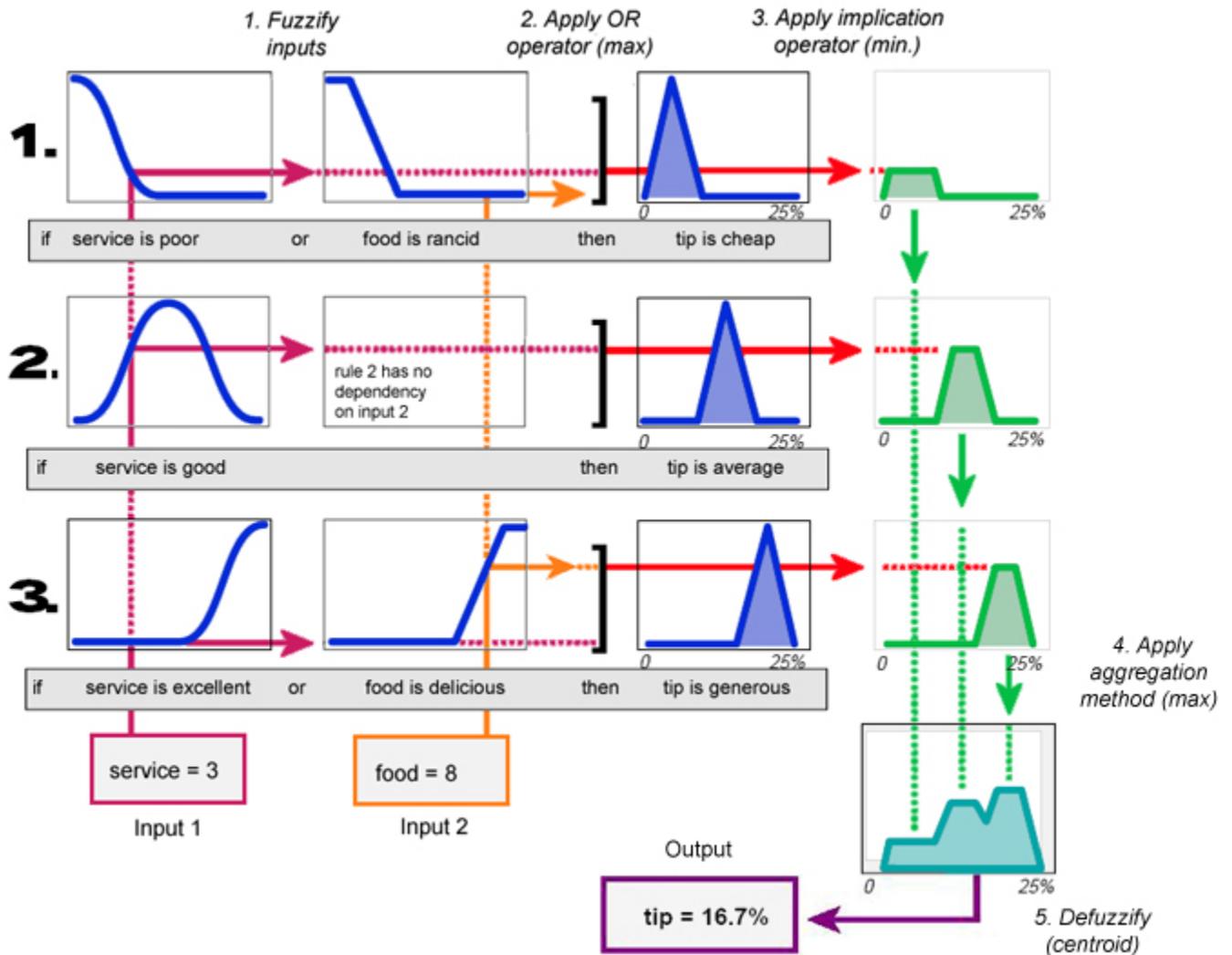


Figura 9 - Imagem retirada de [17]. Todas as etapas do método de Mamdani

Vale ressaltar que o método de Mamdani apresenta um melhor desempenho quando há um número pequeno de variáveis, pois a medida que a quantidade de variáveis aumenta, o número de regras irá crescer exponencialmente e quanto mais regras forem geradas mais difícil será adequar elas a um problema específico. Se o número de variáveis do antecedente for muito grande, será difícil compreender a

relação dos antecedentes com os consequentes. Portanto, a construção de novas regras se tornará mais difícil.

2.6 NetLogo

NetLogo é um ambiente de modelagem programável para simular fenômenos naturais e sociais. Foi escrito por Uri Wilensky em 1999 e está em desenvolvimento contínuo desde então pelo Center for Connected Learning and Computer-Based Modeling. [5]

A ferramenta NetLogo é particularmente adequada para modelar sistemas complexos que evoluem com o passar do tempo. Os programadores podem dar instruções a centenas ou milhares de agentes que operam de forma independente. Isto torna possível explorar a conexão entre o comportamento de indivíduos e os padrões que emergem da interação de muitos indivíduos. [5]

O NetLogo possui uma extensa documentação e tutoriais. Ele também vem com uma biblioteca de modelos, que é uma grande coleção de simulações pré-escritas que podem ser usadas e modificadas. Estas simulações abrangem muitas áreas de conteúdo, desde ciências naturais e sociais, incluindo a biologia e a medicina, física e química, matemática e ciência da computação, economia e psicologia social. [5]

"Apesar da linguagem NetLogo ser bastante fácil de aprender, não põe limites à sofisticação dos modelos que podem ser criados. É particularmente bem adaptada à modelação de sistemas compostos de indivíduos autônomos que interagem entre si."

[14]

O NetLogo permite que seja possível implementar um cenário específico de forma muito simples, pois ele funciona basicamente com dois tipos de objectos, *patches*, que equivalem as casas de um tabuleiro de xadrez , e *turtles* ou agentes, que equivalem a um peão. O cenário pode ter qualquer dimensão NxM e é possível ter qualquer número de *turtles*.

Todo modelo inicia-se com um cenário e um número de agentes, e o modelo vai mudando a medida que o tempo passa e os agentes interagem entre si e também com os *patches*, de acordo com a implementação utilizada para o comportamento delas.

3. Desenvolvimento

Neste trabalho foi criado um modelo de simulação de agentes onde cada um deles é responsável por suas ações e leva em consideração agentes e obstáculos presentes dentro de sua área de percepção.

3.1 Modelo proposto

O modelo desenvolvido nesse trabalho utilizou como base os modelos de Reynolds, na parte sobre comportamento em relação aos outros agentes, [10][11][12] e [2], na detecção de colisões e desvio com outros agentes e obstáculos. A principal diferença foi a utilização de lógica nebulosa na tomada de decisões dos agentes no lugar de formulas complexas utilizados em outros trabalhos.

A implementação do comportamento dos agentes em um grupo se divide em duas etapas principais: desviar de obstáculos e agrupamento. Nelas é levada em consideração a área de percepção que é definida por um cone (Figura 10).



Figura 10 – Cone com raio 5 e ângulo 60

Para verificar a influência de um agente ou obstáculo na decisão do agente utiliza-se Lógica Nebulosa na área de percepção. É verificado o valor da distância nos conjuntos *distância-perto*, *distância-médio* e *distância-longe*. Também é verificado o valor para o ângulo nos conjuntos *ângulo-pequeno*, *ângulo-médio* e *ângulo-grande*. Conforme mostrado na Figura 11.

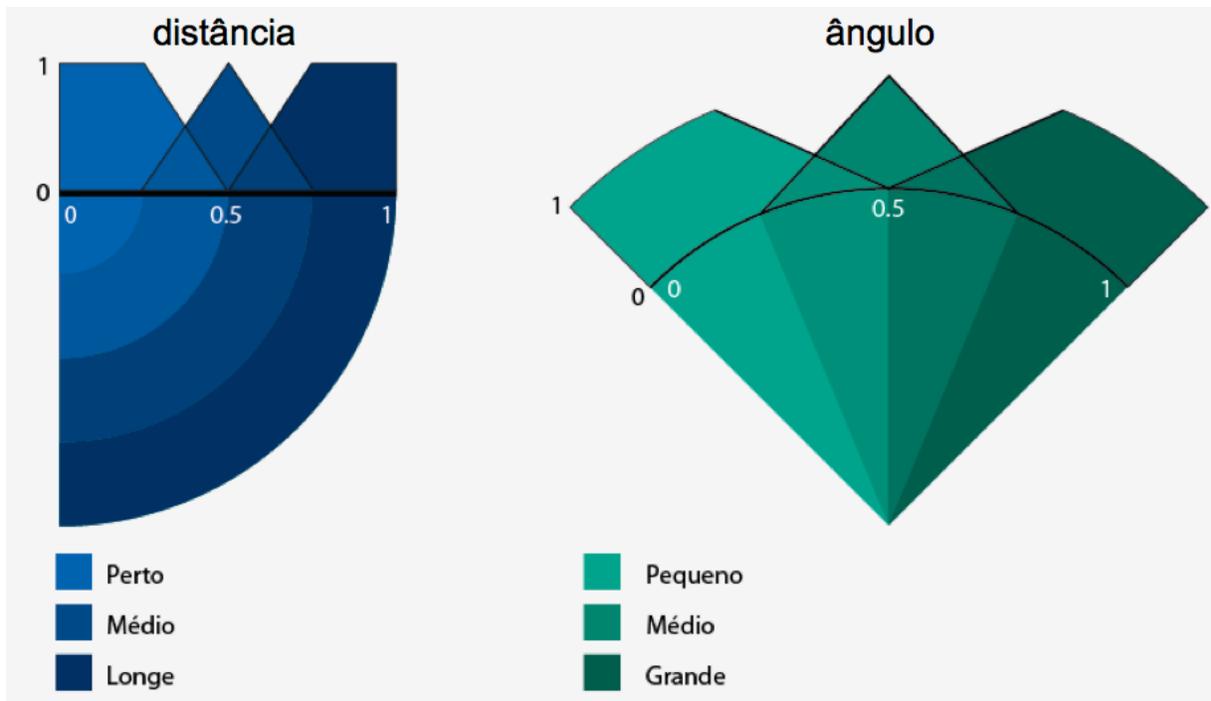


Figura 11 – Conjuntos utilizados na Lógica Nebulosa.

A relação da distância com o ângulo implicará em quantos graus o agente deve ajustar sua direção, a partir disso foram criadas as regras de inferência, que são mostradas na Figura 12.

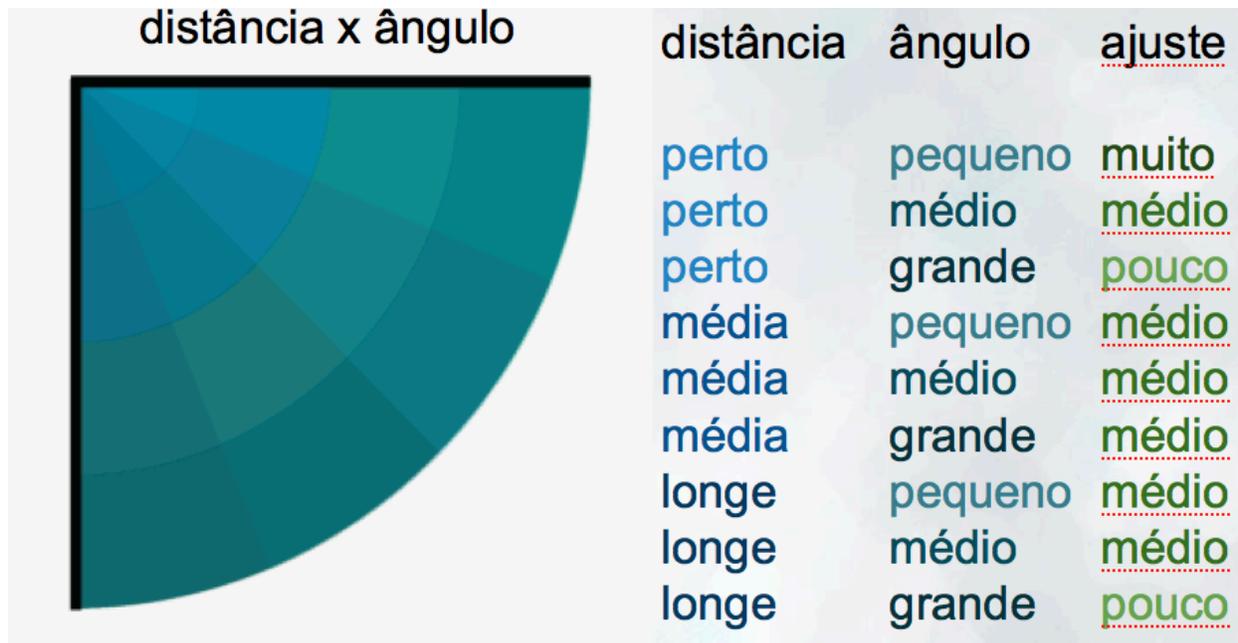


Figura 12 – Regras de inferência do modelo proposto.

Finalmente, o algoritmo do comportamento dos agente funciona da seguinte forma: Primeiramente verifica-se a existência de algum obstáculo na área de percepção de obstáculos do agente. Levando em consideração a distância do obstáculo e o ângulo do agente em relação ao obstáculo, utiliza-se lógica nebulosa para calcular em quantos graus o agente deve ajustar sua direção para desviar do obstáculo. São calculados os valores da distância nos conjuntos *distância-perto*, *distância-médio* e *distância-longe* assim como para o ângulo nos conjuntos *ângulo-pequeno*, *ângulo-médio* e *ângulo-grande*. Para o conseqüente são utilizados os conjuntos *ajuste-pouco*, *ajuste-médio* e *ajuste-muito*. A seguir aplica-se o operador *OR* para obter o maior valor entre as combinações de cada conjunto de distância com

os conjuntos de ângulo, esse valor é utilizado no conseqüente da regra. Por exemplo, se o ângulo se encontra no conjunto *ângulo-pequeno* e a distância no conjunto *distância-média*, no conseqüente, *ajuste-médio*, será utilizado o maior valor entre os dois conjuntos. Após a agregação de todos os valores dos conseqüentes será feita a defuzzyficação para se obter o valor de quantos graus o agente deve virar para desviar do obstáculo utilizando o método do centróide.

Caso não seja detectado nenhum obstáculo é chamado o método de agrupamento. O agrupamento se divide em três etapas que são baseadas no grupo de agentes presentes na área de percepção do agente: afastar-se do grupo, alinhar-se em relação à direção média do grupo e mover-se em direção ao centro do grupo.

Inicialmente verifica-se a existência de outros agentes dentro de um cone de visão, esta área será a sendo a área de percepção do agente, o comportamento de agrupamento será baseado nos agentes que estiverem dentro dessa área. Uma vez descoberto quais agentes estão no seu campo de percepção inicia-se a etapa de afastamento, calcula-se a diferença entre a direção que o agente está se movendo e o vetor direção contrário à média das posições dos agentes e também a distância do agente em relação média das posições. Calcula-se a partir desses valores quantos graus o agente deverá virar para se afastar do grupo utilizando lógica nebulosa como na etapa de desviar de obstáculos. Essa etapa é mostrada na Figura 12.

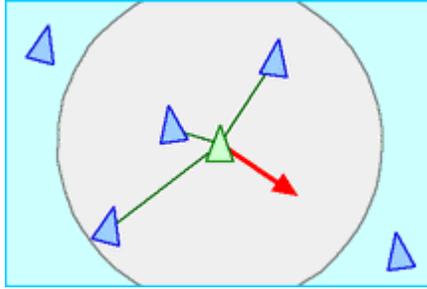


Figura 12 – Afastar-se do grupo. Adaptada de [20]

Como o agente mudou sua direção na etapa anterior é verificado novamente quem são os vizinhos dele. Calcula-se a direção média dos agentes e verifica-se quão diferente da direção do agente ela é e utiliza-se lógica nebulosa para calcular quantos graus o agente deve virar para que se alinhe com o grupo. Nessa etapa a resposta depende somente da direção média dos vizinhos, portanto se a diferença for pequena o agente virará poucos graus, se for média virará um valor mediano de graus e se for grande virará muitos graus. Essa etapa é representada pela Figura 20

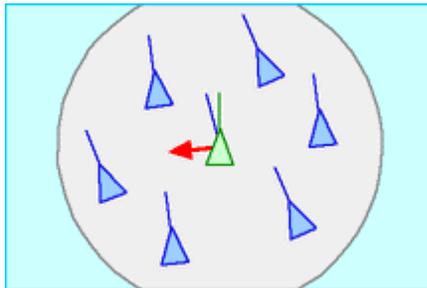


Figura 13 – Alinhar-se em relação ao grupo. Adaptada de [20]

Na terceira etapa calcula-se verifica-se novamente o grupo dos vizinhos e agora o valor de quantos graus devem ser virados é calculado em relação à diferença da

direção do centro do grupo em relação a direção atual do agente e de sua distância em relação a posição média do grupo. Essa etapa é resumida na Figura 14.

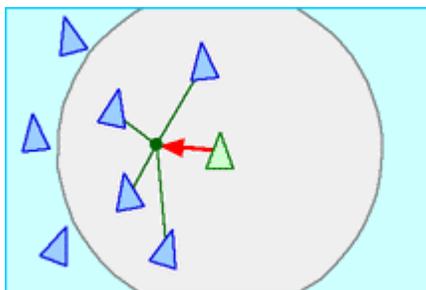


Figura 14 – Mover-se em direção ao centro do grupo. Adaptada de [20]

Todos os conjuntos nebulosos são equivalentes ao presente na Figura 15, sendo que o eixo x da figura representa a porcentagem da variável nebulosa em questão, no Netlogo define-se o valor máximo da variável, 10 por exemplo, então os valores que serão utilizados para a construção dos conjuntos nebulosos serão 0, 2.5, 5, 7.5 e 10.

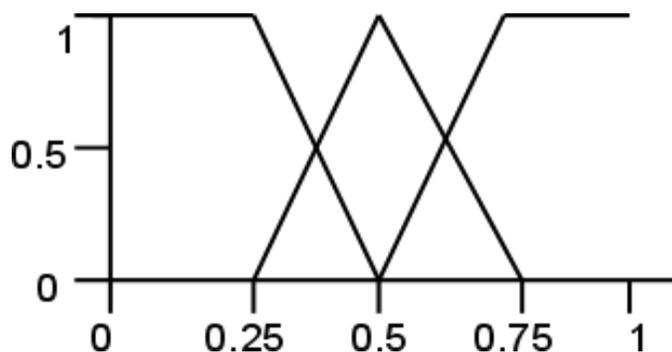


Figura 15: Conjunto nebuloso genérico

4. Validação

Foram realizadas algumas simulações para que se pudesse avaliar o resultado do modelo proposto.

Três cenários retirados de [18] e [19] foram replicados no NetLogo para verificar quão próximo destes outros trabalhos o modelo proposto se parece. Nesses cenários os *patches* azuis representam paredes ou obstáculos e os círculos são os agentes. Primeiro é mostrado o experimento original depois seu similar no NetLogo. As imagens mostradas foram capturadas após algum tempo de simulação para que fosse observado o efeito de agrupamento dos agentes.

Os agentes possuem um ciclo de ações que são repetidas a cada iteração da simulação, conforme explicado no Modelo Proposto.

4.1 Resultados obtidos

A seguir mostram-se os cenários que foram simulados no NetLogo utilizando o modelo proposto.

4.1.1 Corredor com alargamento

Esse cenário, retirado de [19], mostra um fenômeno causado pelo comportamento responsável pela formação de um grupo, como os agentes tentam se manter a uma distância segura de outros agentes eles acabam se espalhando em corredores que possuem áreas onde ficam mais largos, em seguida eles se encontram

numa situação de afunilamento (Figura 16).

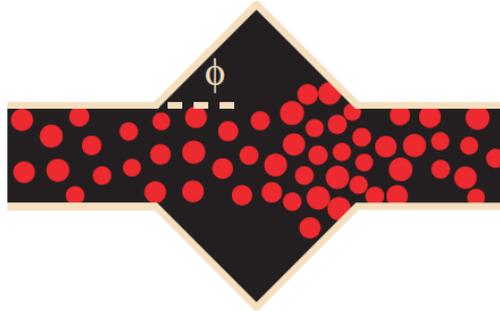


Figura 16 - Imagem retirada de [19] que mostra o resultado de uma multidão passando por um corredor que possui uma área onde ele se alarga.

Nesta simulação observou-se um comportamento muito parecido com o esperado, a medida que os agente chegavam na parte mais alargada do corredor eles foram se afastando devido ao comportamento de separação no *flocking*, em seguida por causa do afunilamento das paredes eles voltaram a se aproximar, devido ao alto número de agentes nessa simulação e ao fato de a prioridade em desviar das paredes ser maior que a de desviar de outros agentes acontecem colisões entre os agentes, conforme pode ser observado na Figura 17.

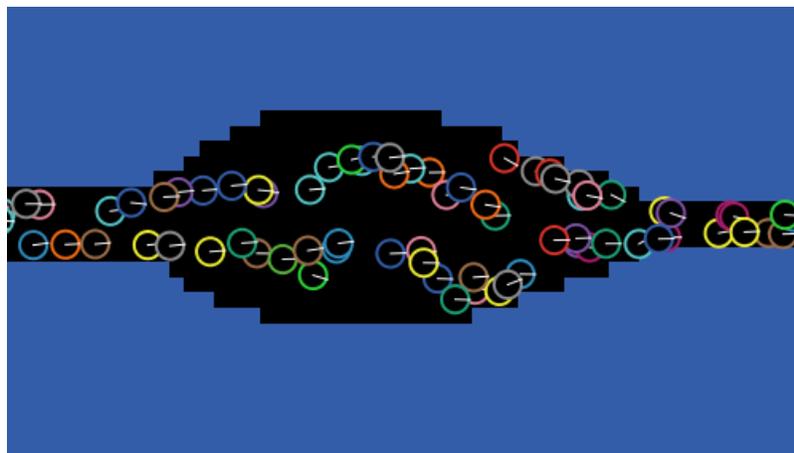


Figura 17 - Resultado obtido no modelo proposto.

4.1.2 Cruzamento

Nesse cenário após alguns instantes de simulação acontece uma explosão fazendo com que os agentes fujam para longe do perigo (Figura 18).



Figura 18 - Imagem retirada de [18] que mostra o resultado da simulação em um cruzamento.

Para simular o cenário acima deixou-se a simulação decorrer por um tempo, assim foi possível disparar o evento de fuga com os agentes já em movimento como se fossem pessoas que estivessem passando por ali e não um grupo que estava reunido no local, como o mapa é contínuo nos lados os agentes que chegavam ao “fim” da tela apareciam do outro lado. Após a detecção do fogo os agentes começaram a mover-se na direção oposta obedecendo o comportamento de *flocking* (Figura 19).

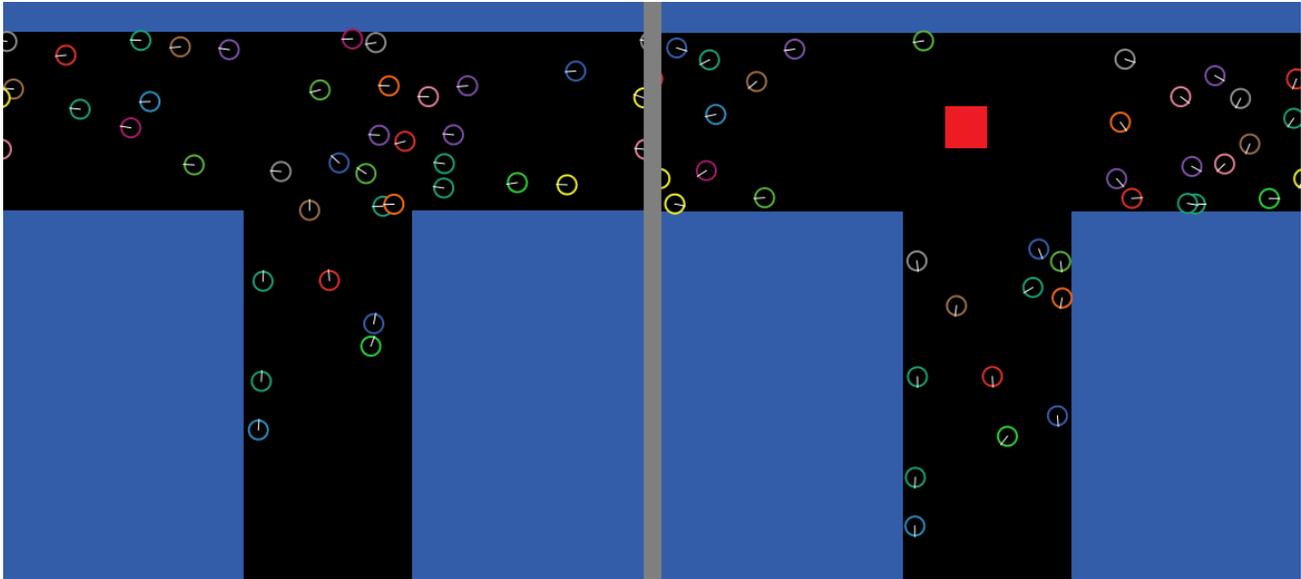


Figura 19 - Resultado da simulação utilizando o modelo proposto

4.1.3 Centro da cidade:

Esse cenário (Figura 20) simula um local no centro de alguma cidade onde o fluxo de pedestre eh grande. Cada círculo representa um agente e o rastro, o caminho percorrido.

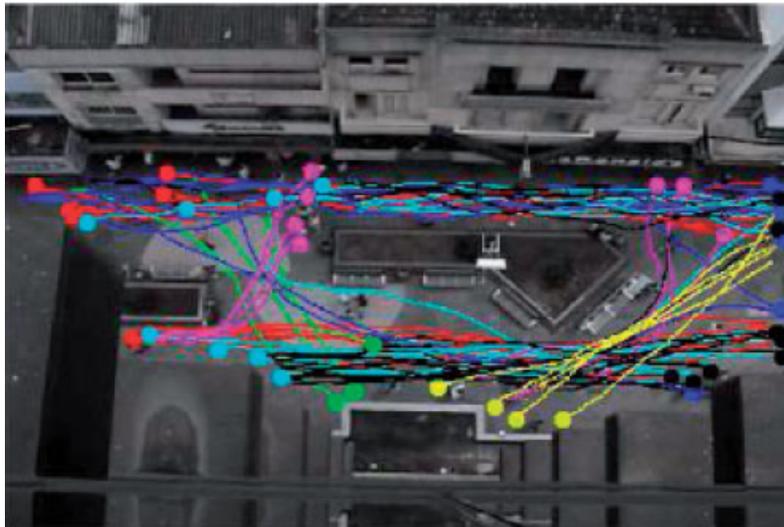


Figura 20 - Imagem retirada de [18] que mostra o resultado da simulação no centro da cidade.

Na simulação representada pela Figura 21 os agentes paravam de andar após alcançarem os *patches* verdes. Pode-se observar que os agentes desviam de sua direção inicial para tentar andar com um grupo e em alguns casos obstáculos os impedem.

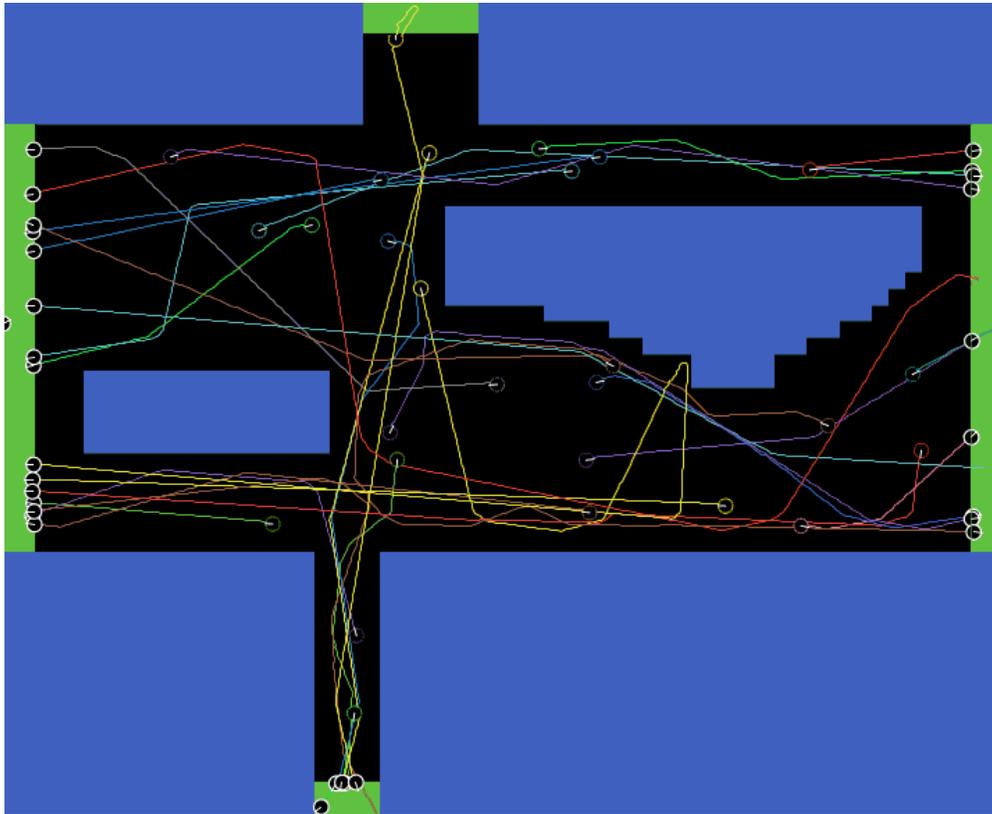


Figura 21 - Resultado da simulação utilizando o modelo proposto

4.1.4 Ambiente sem obstáculos:

Nesse cenário é possível observar mais facilmente como o comportamento de *flocking* funciona, pois não existe a intervenção de obstáculos. Nesse experimento foram colocados cinquenta agentes, com posição e a direção iniciais aleatórias, em um ambiente com duzentos *patches* de altura e duzentos *patches* de largura. A Figura 22 mostra os estados dessa simulação.

Pode-se observar pelas outras figuras que no decorrer da simulação os agentes começam a formar grupos. Alguns agentes acabam se separando de um grupo e indo para outro devido à sua área de percepção, num determinado momento havia mais agentes indo em uma direção do que outra então esse agente acabou trocando de grupo, apesar disso a tendência é que com o passar do tempo os grupos se agrupem formando um grande grupo, como mostra a Figura 23.

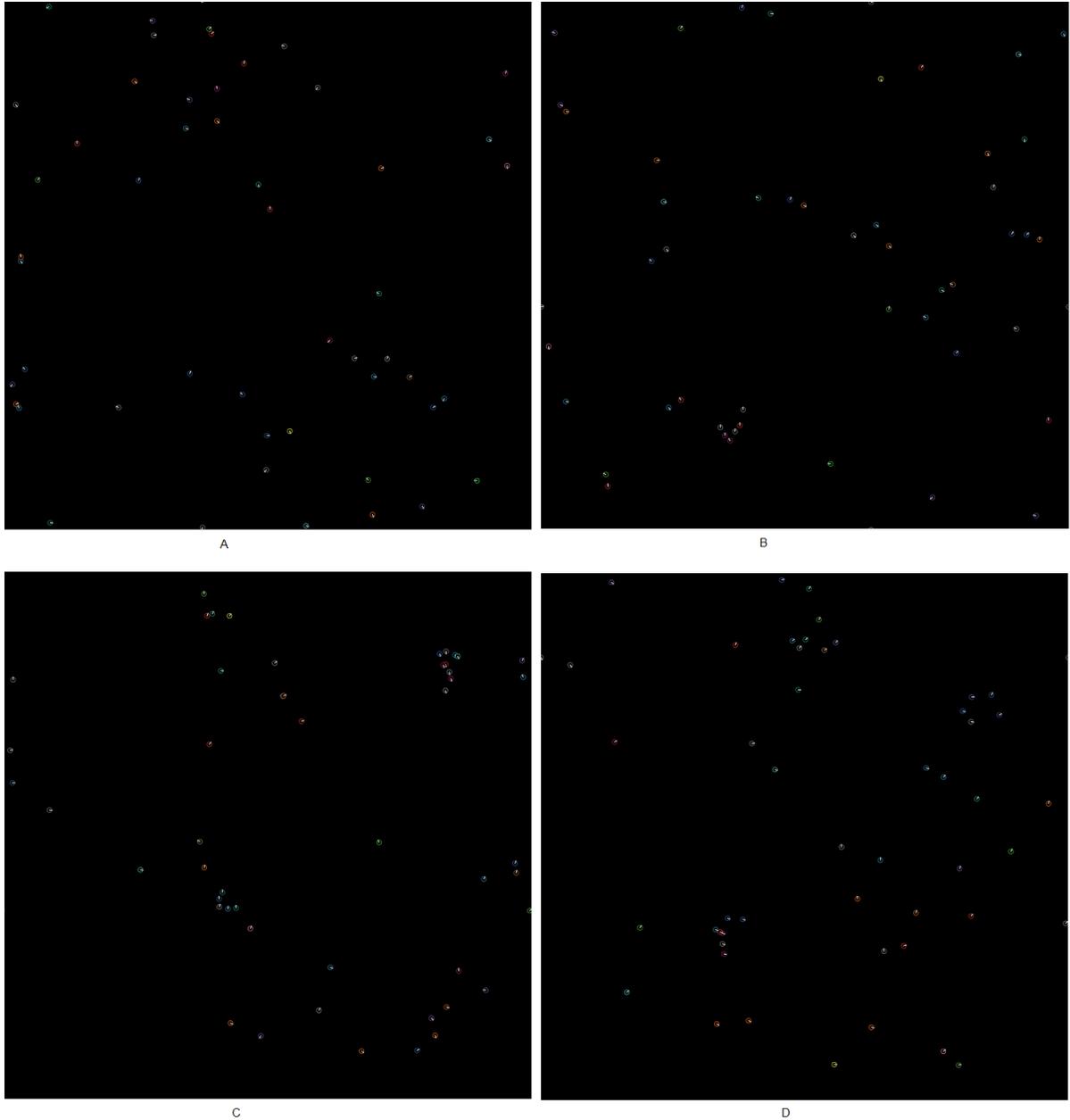


Figura 22 - Estados da simulação sem obstáculos.

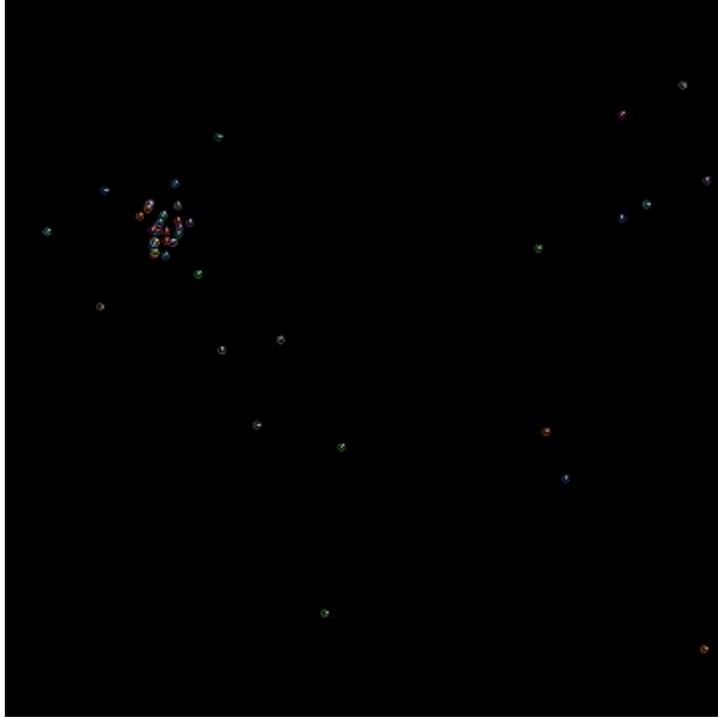


Figura 23 - Formação de um grande grupo após algum tempo de simulação

5. Conclusão

A partir dos resultados obtidos através das simulações pode-se afirmar que os objetivos foram alcançados, o modelo proposto gera resultados semelhantes aos encontrados em outros modelos, onde os agentes formam grupos que tendem a se mover em uma mesma direção ao mesmo tempo que evitam, na medida do possível, colisões com obstáculos e outros agentes. Em outros modelos as colisões são totalmente evitadas o que não condiz com a realidade, em situações de evacuação pessoas acabam se chocando, nesse modelo essas colisões acontecem.

Verificou-se que o modelo proposto gerou um resultado que representa de forma mais real a reação de pessoas, uma vez que com o uso de Lógica Nebulosa foi possível simular de forma mais natural aos agentes, e que com a área de percepção o agente só considera, em sua decisão sobre qual direção ir, os agentes e obstáculos que estão próximos.

5.1 Trabalhos futuros

Implementar o modelo proposto em um simulador que permita a criação de mapas mais complexos onde seja possível gerar um modelo tridimensional de locais com alto fluxo de multidões.

Acrescentar ao modelo fatores sócio-psicológicos como medo, altruísmo, dependência, entre outros.

Adicionar tratamento para colisões que não puderam ser evitadas, como pessoas caindo e se machucando.

Referências bibliográficas

- [1] BRAUN, MUSSE, BARDO e BADMANN, Modeling Individual Behaviors in Crowd Simulation
- [2] CONY, BIXO, JUNG, MAGALHÃES e MUSSE, A Perceptive Model for Virtual Agents in Crowds.
- [3] PAIVA, SILVA e LOBO NETTO, Um Simulador de Multidões em Situações de Emergência seguindo Recomendações da CIPA.
- [4] THIRION e BASU, Real-Time Crowd Simulation for Emergency Planning
- [5] Wilensky, U. (1999). NetLogo. HYPERLINK "<http://ccl.northwestern.edu/netlogo/>" <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
- [6] ZADEH, Fuzzy Logic = Computing with Words. IEEE Transactions on fuzzy systems, vol 4, no.2, Mar 1996.
- [7] DIRK HELBING, LUBOS BUZNA, ANDERS JOHANSSON e TORSTEN WERNER, Self-Organized Pedestrian Crowd Dynamics: Experiments, Simulations, and Design Solutions
- [8] D.A. PURSER, M. BENSILUM, Quantification of behavior for engineering design standards and escape time calculations
- [9] XIAOSHAN PAN, CHARLES S. HAN, KEN DAUBER, KINCHO H. LAW, Human and social behavior in computational modeling and analysis of egress
- [10] SHARAD SHAMA, Simulation of Agent Behavior in a Goal Finding Application

- [11] SHARAD SHAMA, Modeling Emergency Scenarios in Virtual Evacuation Environment
- [12] SHARAD SHARMA, HARPREET SINGH e ATUL PRAKASH, Multi-Agent Modeling and Simulation of Human Behavior in Aircraft Evacuations
- [13] BIN ZH, TAO LIU e YONGCHUAN TANG, Research on Pedestrian Evacuation Simulation Based on Fuzzy Logic
- [14] <http://cftc.cii.fc.ul.pt/PRISMA/capitulos/netlogo/>
- [15] REYNOLDS, CRAIG W. Flocks, Herds, and Schools: A Distributed Behavioral Model
- [16] SANDRi, SANDRA, CORREA, CLÁUDIO. Lógica Nebulosa
- [17] http://www.dma.fi.upm.es/java/fuzzy/fuzzyinf/mamdani_en.htm.
- [18] MUSSE, JUNG, JACQUES, BRAUN, Using computer vision to simulate the motion of virtual agents
- [19] HELBING, FARKAS e VICSEK, Simulating dynamical features of escape panic
- [20] www.red3d.com/cwr/boids/

Apêndices

Apêndice 1 - Artigo

Modelo para Simulação de Multidões Utilizando Lógica Nebulosa

Tharak Zuchetto Nasser¹

¹Ciências da Computação – Universidade Federal de Santa Catarina (UFSC)
Campus Universitário Reitor João David Ferreira Lima Trindade - Florianópolis - Santa Catarina
- Brasil CEP: 88040-970

Abstract. Simulations of human behavior in panic situations are used in architectural planning environments where there is a large flow of people and from the results generated by these can foresee with some degree of realism, as a crowd would behave in the event of an emergency , allowing the location to be improved to facilitate the evacuation of it.

This work proposes to develop a simulation model that mimics the behavior of people in a more realistic way with the use of fuzzy logic instead of the mathematical formulas that define the behavior of the agents that represent humans.

Resumo. Simulações do comportamento humano em situações de pânico são utilizadas no planejamento arquitetônico de ambientes onde existe um grande fluxo de pessoas e, a partir dos resultados gerados por elas, é possível prever, com certo grau de realismo, como uma multidão se comportaria no caso de haver necessidade de evacuar o local e possibilitando que seja feita uma melhoria no local para facilitar a evacuação do mesmo.

Esse trabalho propõe desenvolver um modelo de simulação que imite o comportamento de pessoas de uma forma mais realista através da utilização de Lógica Nebulosa no lugar das fórmulas matemáticas que definem o comportamento dos agentes que representam seres humanos.

1. Introdução

Em simulações de comportamentos de multidões são utilizadas fórmulas complexas na implementação do comportamento dos agentes. E essas implementações são baseadas em fenômenos físicos, como soma dos vetores das forças que estão sendo aplicadas em um agente. A ideia desse trabalho é que se utilize lógica nebulosa para substituir essas regras, uma vez que acredita-se que, com lógica, nebulosa é mais fácil de simular o comportamento de um ser humano.

Na lógica nebulosa não se usam valores fixos para representar um valor. É criado um conjunto para representá-lo, por exemplo, quando se pergunta qual a distância entre dois pontos sem a utilização de lógica nebulosa a resposta será em metros, já com lógica nebulosa será distância curta, distância média ou distância longa.

Seres humanos utilizam uma linguagem natural e percepções mentais para decidir que ações vão tomar, chegando a conclusões expressas em palavras. Estas não possuem um significado exato, assim como as palavras usadas na representação de um sistema nebuloso. Portanto, acredita-se que, utilizando lógica nebulosa, será possível obter um resultado mais próximo ao real.

2. Metodologia

Este artigo está dividido em quatro etapas.

- Etapa 1 - Fundamentação Teórica: Foi feito um estudo sobre modelos que são utilizados em simulações de comportamento de pessoas em multidões, e também de modelos para simular o comportamento dessas multidões em situações de pânico. Também foi feita uma pesquisa sobre lógica nebulosa e suas aplicações, ao final dessa pesquisa é explicado o método que melhor se aplicou ao modelo proposto neste trabalho. No final dessa etapa é falado sobre o programa que foi utilizado para implementar o modelo proposto.
- Etapa 2 - Desenvolvimento: Nesta etapa é apresentado o modelo proposto que foi criado a partir das pesquisas feitas na etapa anterior. Em seguida é explicado como

ocorreram as simulações, assim como seus parâmetros. Ao final são mostrados os cenários que foram utilizados para testar a eficiência do modelo proposto.

- Etapa 3 - Validação: Primeiramente são definidos os parâmetros da avaliação que decidirá se o modelo proposto atinge seus objetivos. Depois são simulados os cenários definidos anteriormente utilizando-se o modelo proposto. Após a coleta de dados é feita a avaliação.
- Etapa 4 - Conclusão: Aqui são mostrados as conclusões geradas a partir dos resultados obtidos nas simulações feitas na etapa anterior

3. Fundamentação Teórica

Neste capítulo serão apresentados os modelos e regras que influenciaram diretamente no desenvolvimento deste trabalho. Primeiramente é mostrado o modelo que serviu como início para este trabalho, o modelo de Helbing. Tendo esse modelo como base foram agregadas características de outros modelos, como área de percepção e detecção e desvio de obstáculos. Foi adicionado também o comportamento de *flocking*, para que os agentes se comportassem de forma mais natural. Após isso os cálculos que existiam no modelo foram substituídos pela Lógica Nebulosa.

3.1. Modelo de Helbing

Helbing [7] propôs um modelo baseado em física e forças sócio-psicológicas para descrever o comportamento de multidões em situações de emergência. Ele utiliza um sistema onde cada partícula i de massa m_i possui uma velocidade v , ou seja, a velocidade com a qual se deslocará em direção ao seu objetivo corresponde ao vetor e_i . Essa velocidade é recalculada a cada intervalo de tempo t_i . Simultaneamente as partículas tendem a manter uma distância de outras entidades, j , e de paredes, w , seus vetores direção são f_{ij} e f_{iw} , respectivamente. O cálculo da nova velocidade no tempo t se dá pela equação da imagem a seguir.[1]

3.2. Modelo de percepção por visão para um agente

O modelo construído em A Perceptive Model for Virtual Agents in Crowds utiliza uma abordagem de zonas para simular a percepção visual do agente. Essa região possui dois atributos, distância, d , e ângulo, a , que são fixados no início da simulação. Outro agente ou um obstáculo será percebido quando entrar nessa região de percepção do agente. [2]

3.3. Detecção e desvio de obstáculos e agentes

O sistema proposto por SHARMA [10][11][12] utiliza um método de detecção colisão com base na distância. Isso envolve o uso de distância entre dois objetos para determinar se

os agentes colidiram. Este método favorece círculos perfeitos, mas pode ser melhorado para outras formas também.

A colisão é verificada pelo agente que esta se movendo em direção ao objetivo. Quando a distância entre o agente e um outro agente fica abaixo de um valor limite, o método responsável por calcular o desvio para não haver uma colisão entre os agentes é chamado. Em [2] é utilizada uma abordagem similar, como o agente fica responsável por verificar eventuais colisões esta é tratada sem muito custo de processamento, diferente do que aconteceria se existisse uma entidade global verificando a cada momento se algum agente está em rota de colisão com outro. Como resultado é gerada uma animação mais suave e natural.

3.4. Flocking

Reynolds[15] sugeriu uma abordagem eficiente, robusta e condizente com a realidade que se baseia na ideia de que o movimento de um grupo, na verdade, é gerado pelo comportamento individual de cada membro do mesmo. Nessa abordagem, simulou-se uma parte do comportamento de um pássaro responsável pelo comportamento que permite que ele faça parte de um bando. Além da parte responsável pela percepção e aspectos físicos do voo.

Em [15] foi usada uma abordagem baseada em sistemas de partículas, onde partícula correspondia a um pássaro. Cada pássaro tentava, independentemente e ao mesmo tempo, ficar junto de seu bando e evitar colisões com seus companheiros e objetos do ambiente.

Para conseguir simular o comportamento de grupos, Reynolds dividiu o comportamento dos agentes em três etapas: evitar colisão, igualar velocidade e ir para o centro do grupo.

No algoritmo para evitar colisão leva-se em consideração a posição dos membros que estão a certa distância, e para igualar a velocidade, a velocidade dos mesmos.

Como cada agente tem uma percepção diferente o centro do grupo, na verdade, é usado o centro dos agentes que estão dentro da área de percepção dele para calcular a direção que ele deve tomar para este centro local. Grupos na vida real se separam quando tentam se desviar de um obstáculo, membros do grupo não se preocupam se nem todos os

membros estão indo na mesma direção, somente os que estão dentro de sua área de percepção.

Esses comportamentos que compõem o modelo estão expressos em questão de "membros próximos", que são os agentes que estão em uma área circular centralizada no agente.

A parte mais interessante da simulação de um grupo, segundo Reynolds, é a interação com outros objetos do ambiente. O comportamento isolado do grupo tende a alcançar uma estabilidade. Por exemplo, os conflitos entre evitar colisão e ir para o centro do grupo não levam a um conflito, mas sim a um equilíbrio. Obstáculos no ambiente e a tentativa do grupo de desviar deles aumentam a aparente complexidade do comportamento do grupo.

Ainda segundo Reynolds, somar um valor ao ângulo do vetor direção do agente é o melhor jeito de simular um agente guiado pela visão pois ele considera somente os obstáculos presentes na sua área de percepção.

3.5. Lógica Nebulosa

De acordo com [6], a principal contribuição da lógica nebulosa é poder programar utilizando palavras. Nenhuma outra metodologia, como, lógica de predicados, teoria probabilística, teoria de redes neurais, redes Bayesianas, etc., servem para esse propósito. Zadeh chama isso de Computing with Words (CW). Claro que essa não é a única utilidade de lógica nebulosa, mas é a que melhor se aplica no contexto deste trabalho.

O papel da lógica nebulosa em CW é propagar as premissas para conclusões. Dada uma entrada x , verifica-se em que conjunto nebuloso ele se encontra. Por exemplo, muito longe, através da lógica nebulosa, será transformado em uma saída, aumentar velocidade.

3.6. NetLogo

NetLogo é um ambiente de modelagem programável para simular fenômenos naturais e sociais. Foi escrito por Uri Wilensky em 1999 e está em desenvolvimento contínuo desde então pelo Center for Connected Learning and Computer-Based Modeling. [5]
A ferramenta NetLogo é particularmente adequada para modelar sistemas complexos que evoluem com o passar do tempo. Os programadores podem dar instruções a centenas ou milhares de agentes que operam de forma independente. Isto torna possível explorar a conexão entre o comportamento de indivíduos e os padrões que emergem da interação de muitos indivíduos. [5]

4. Desenvolvimento

Neste trabalho foi criado um modelo de simulação de agentes onde cada um deles é responsável por suas ações e leva em consideração agentes e obstáculos presentes dentro de sua área de percepção.

4.1. Modelo Proposto

O modelo desenvolvido nesse trabalho utilizou como base os modelos de Reynolds, na parte sobre comportamento em relação aos outros agentes, [10][11][12] e [2], na detecção de colisões e desvio com outros agentes e obstáculos. A principal diferença foi a utilização de lógica nebulosa na tomada de decisões dos agentes no lugar de formulas complexas utilizados em outros trabalhos.

A implementação do comportamento dos agentes em um grupo se divide em duas etapas principais: desviar de obstáculos e agrupamento. Nelas é levada em consideração a área de percepção que é definida por um cone.

Para verificar a influência de um agente ou obstáculo na decisão do agente utiliza-se Lógica Nebulosa na área de percepção. É verificado o valor da distância nos conjuntos distância-perto, distância-médio e distância-longe. Também é verificado o valor para o ângulo

nos conjuntos ângulo-pequeno, ângulo-médio e ângulo-grande.

A relação da distância com o ângulo implicará em quantos graus o agente deve ajustar sua direção, a partir disso foram criadas as regras de inferência

Finalmente, o algoritmo do comportamento dos agente funciona da seguinte forma: Primeiramente verifica-se a existência de algum obstáculo na área de percepção de obstáculos do agente. Levando em consideração a distância do obstáculo e o ângulo do agente em relação ao obstáculo, utiliza-se lógica nebulosa para calcular em quantos graus o agente deve ajustar sua direção para desviar do obstáculo. São calculados os valores da distância nos conjuntos distância-perto, distância-médio e distância-longe assim como para o ângulo nos conjuntos ângulo-pequeno, ângulo-médio e ângulo-grande. Para o conseqüente são utilizados os conjuntos ajuste-pouco, ajuste-médio e ajuste-muito. A seguir aplica-se o operador OR para obter o maior valor entre as combinações de cada conjunto de distância com os conjuntos de ângulo, esse valor é utilizado no conseqüente da regra. Por exemplo, se o ângulo se encontra no conjunto ângulo-pequeno e a distância no conjunto distância-média, no conseqüente, ajuste-médio, será utilizado o maior valor entre os dois conjuntos. Após a agregação de todos os valores dos conseqüentes será feita a defuzzyficação para se obter o valor de quantos graus o agente deve virar para desviar do obstáculo utilizando o método do centróide.

Caso não seja detectado nenhum obstáculo é chamado o método de agrupamento. O agrupamento se divide em três etapas que são baseadas no grupo de agentes presentes na área de percepção do agente: afastar-se do grupo, alinhar-se em relação à direção média do grupo e mover-se em direção ao centro do grupo.

Inicialmente verifica-se a existência de outros agentes dentro de um cone de visão, esta área será a sendo a área de percepção do agente, o comportamento de agrupamento será

baseado nos agentes que estiverem dentro dessa área. Uma vez descoberto quais agentes estão no seu campo de percepção inicia-se a etapa de afastamento, calcula-se a diferença entre a direção que o agente está se movendo e o vetor direção contrário à média das posições dos agentes e também a distância do agente em relação média das posições. Calcula-se a partir desses valores quantos graus o agente deverá virar para se afastar do grupo utilizando lógica nebulosa como na etapa de desviar de obstáculos.

Como o agente mudou sua direção na etapa anterior é verificado novamente quem são os vizinhos dele. Calcula-se a direção média dos agentes e verifica-se quão diferente da direção do agente ela é e utiliza-se lógica nebulosa para calcular quantos graus o agente deve virar para que se alinhe com o grupo. Nessa etapa a resposta depende somente da direção média dos vizinhos, portanto se a diferença for pequena o agente virará poucos graus, se for média virará um valor mediano de graus e se for grande virará muitos graus.

Na terceira etapa calcula-se verifica-se novamente o grupo dos vizinhos e agora o valor de quantos graus devem ser virados é calculado em relação à diferença da direção do centro do grupo em relação a direção atual do agente e de sua distância em relação a posição média do grupo.

5. Validação

Foram realizadas algumas simulações para que se pudesse avaliar o resultado do modelo proposto.

Três cenários retirados de [18] e [19] foram replicados no NetLogo para verificar quão próximo destes outros trabalhos o modelo proposto se parece. Nesses cenários os patches azuis representam paredes ou obstáculos e os círculos são os agentes. Primeiro é mostrado o experimento original depois seu similar no NetLogo. As imagens mostradas foram capturadas após algum tempo de simulação para que fosse observado o efeito de agrupamento dos agentes.

Os agentes possuem um ciclo de ações que são repetidas a cada iteração da simulação, conforme explicado no Modelo Proposto.

4.1 Resultados obtidos

A seguir mostram-se os cenários que foram simulados no NetLogo utilizando o modelo proposto.

4.1.1 Corredor com alargamento

Esse cenário, retirado de [19], mostra um fenômeno causado pelo comportamento responsável pela formação de um grupo, como os agentes tentam se manter a uma distância segura de outros agentes eles acabam se espalhando em corredores que possuem áreas onde ficam mais largos, em seguida eles se encontram numa situação de afunilamento (Figura 1).

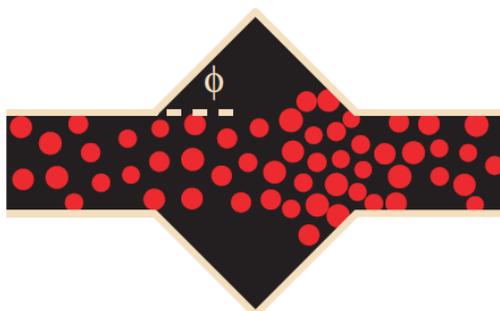


Figura 1 - Imagem retirada de [19] que mostra o resultado de uma multidão passando por um corredor que possui uma área onde ele se alarga.

Nesta simulação observou-se um comportamento muito parecido com o esperado, a medida que os agente chegavam na parte mais alargada do corredor eles foram se afastando devido ao comportamento de separação no flocking, em seguida por causa do afunilamento das paredes eles voltaram a se aproximar, devido ao alto número de agentes nessa simulação e ao fato de a prioridade em desviar das paredes ser maior que a de desviar de outros agentes acontecem colisões entre os agentes, conforme pode ser observado na Figura 2.

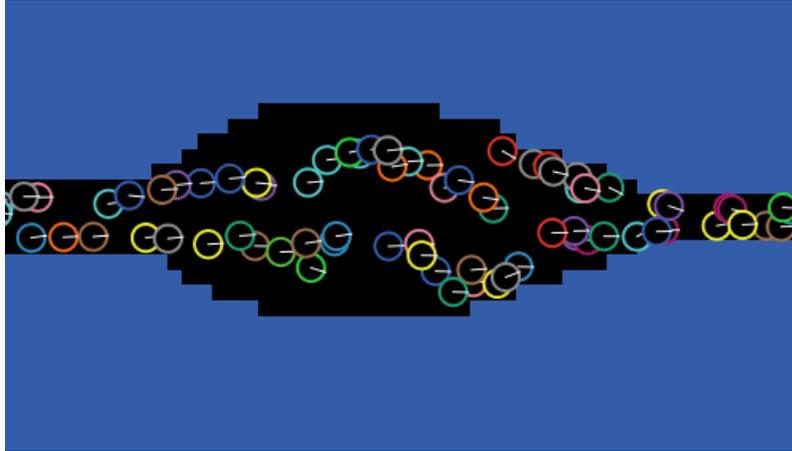


Figura 2 - Resultado obtido no modelo proposto.

4.1.2 Cruzamento

Nesse cenário após alguns instantes de simulação acontece uma explosão fazendo com o que os agentes fujam para longe do perigo (Figura 3).



Figura 3 - Imagem retirada de [18] que mostra o resultado da simulação em um cruzamento.

Para simular o cenário acima deixou-se a simulação decorrer por um tempo, assim foi possível disparar o evento de fuga com os agentes já em movimento como se fossem pessoas que estivessem passando por ali e não um grupo que estava reunido no local, como o mapa é contínuo nos lados os agentes que chegavam ao “fim” da tela apareciam do outro lado. Após a detecção do fogo os agentes começaram a mover-se na direção oposta obedecendo o comportamento de flocking (Figura 4).

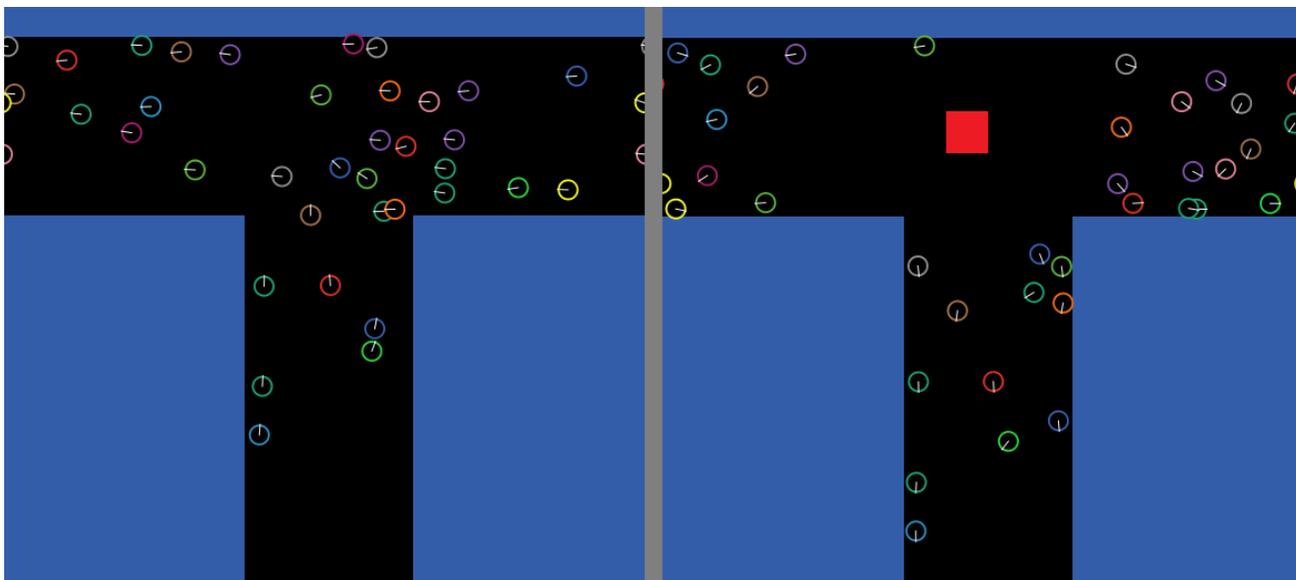


Figura 4 - Resultado da simulação utilizando o modelo proposto

4.1.3 Centro da cidade:

Esse cenário (Figura 5) simula um local no centro de alguma cidade onde o fluxo de pedestre eh grande. Cada círculo representa um agente e o rastro, o caminho percorrido.

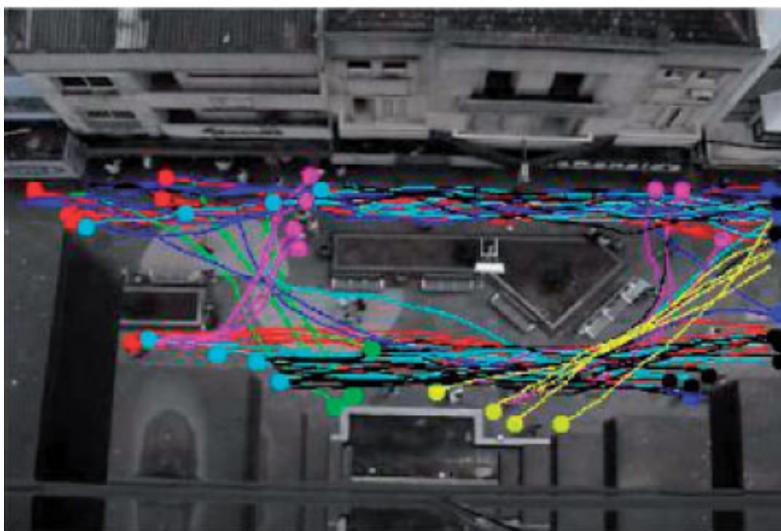


Figura 5 - Imagem retirada de [18] que mostra o resultado da simulação no centro da cidade

Na simulação representada pela Figura 6 os agentes paravam de andar após alcançarem os patches verdes. Pode-se observar que os agentes desviam de sua direção inicial para tentar andar com um grupo e em alguns casos obstáculos os impedem.



Figura 6 - Resultado da simulação utilizando o modelo proposto

4.1.4 Ambiente sem obstáculos:

Nesse cenário é possível observar mais facilmente como o comportamento de flocking funciona, pois não existe a intervenção de obstáculos. Nesse experimento foram colocados cinquenta agentes, com posição e a direção iniciais aleatórias, em um ambiente com duzentos patches de altura e duzentos patches de largura. A Figura 7 mostra os estados dessa simulação.

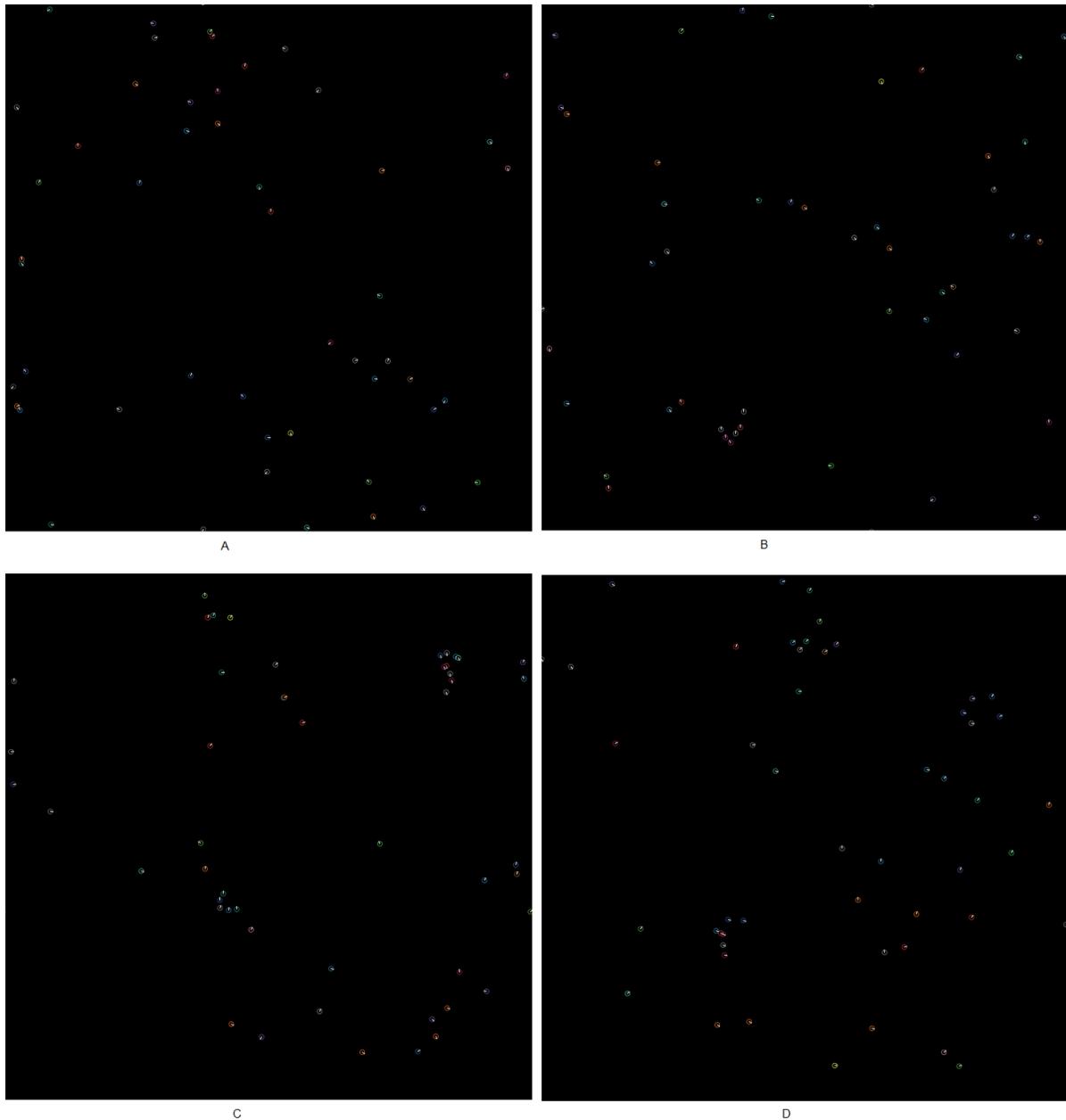


Figura 7 - Estados da simulação sem obstáculos.

Pode-se observar pelas outras figuras que no decorrer da simulação os agentes começam a formar grupos. Alguns agentes acabam se separando de um grupo e indo para outro devido à sua área de percepção, num determinado momento havia mais agentes indo em uma direção do que outra então esse agente acabou trocando de grupo, apesar disso a tendência é que com o passar do tempo os grupos se agrupem formando um grande grupo, como mostra a Figura 8.

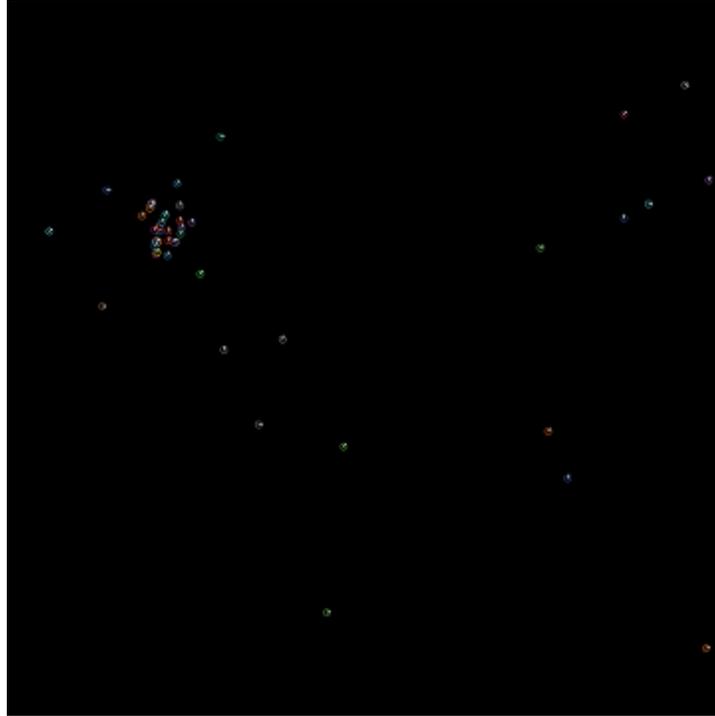


Figura 8- Formação de um grande grupo após algum tempo de simulação

5. Conclusão

A partir dos resultados obtidos através das simulações pode-se afirmar que os objetivos foram alcançados, o modelo proposto gera resultados semelhantes aos encontrados em outros modelos, onde os agentes formam grupos que tendem a se mover em uma mesma direção ao mesmo tempo que evitam, na medida do possível, colisões com obstáculos e outros agentes. Em outros modelos as colisões são totalmente evitadas o que não condiz com a realidade, em situações de evacuação pessoas acabam se chocando, nesse modelo essas colisões acontecem.

Verificou-se que o modelo proposto gerou um resultado que representa de forma mais real a reação de pessoas, uma vez que com o uso de Lógica Nebulosa foi possível simular de forma mais natural aos agentes, e que com a área de percepção o agente só considera, em sua decisão sobre qual direção ir, os agentes e obstáculos que estão próximos.

7. References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in brackets, e.g. [Knuth 1984], [Boulic and Renault 1991]; or dates in parentheses, e.g. Knuth (1984), Smith and Jones (1999).

The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the subsequent should be indented by 0.5 cm.

References

- [1] BRAUN, MUSSE, BARDO e BADMANN, Modeling Individual Behaviors in Crowd Simulation
- [2] CONY, BIXO, JUNG, MAGALHÃES e MUSSE, A Perceptive Model for Virtual Agents in Crowds.
- [3] PAIVA, SILVA e LOBO NETTO, Um Simulador de Multidões em Situações de Emergência seguindo Recomendações da CIPA.
- [4] THIRION e BASU, Real-Time Crowd Simulation for Emergency Planning
- [5] Wilensky, U. (1999). NetLogo. HYPERLINK "<http://ccl.northwestern.edu/netlogo/>" <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
- [6] ZADEH, Fuzzy Logic = Computing with Words. IEEE Transactions on fuzzy systems, vol 4, no.2, Mar 1996.
- [7] DIRK HELBING, LUBOS BUZNA, ANDERS JOHANSSON e TORSTEN WERNER, Self-Organized Pedestrian Crowd Dynamics: Experiments, Simulations, and Design Solutions
- [8] D.A. PURSER, M. BENSILUM, Quantification of behavior for engineering design standards and escape time calculations

- [9] XIAOSHAN PAN, CHARLES S. HAN, KEN DAUBER, KINCHO H. LAW, Human and social behavior in computational modeling and analysis of egress
- [10] SHARAD SHAMA, Simulation of Agent Behavior in a Goal Finding Application
- [11] SHARAD SHAMA, Modeling Emergency Scenarios in Virtual Evacuation Environment
- [12] SHARAD SHARMA, HARPREET SINGH e ATUL PRAKASH, Multi-Agent Modeling and Simulation of Human Behavior in Aircraft Evacuations
- [13] BIN ZH, TAO LIU e YONGCHUAN TANG, Research on Pedestrian Evacuation Simulation Based on Fuzzy Logic
- [14] <http://cftc.cii.fc.ul.pt/PRISMA/capitulos/netlogo/>
- [15] REYNOLDS, CRAIG W. Flocks, Herds, and Schools: A Distributed Behavioral Model
- [16] SANDRi, SANDRA, CORREA, CLÁUDIO. Lógica Nebulosa
- [17] http://www.dma.fi.upm.es/java/fuzzy/fuzzyinf/mamdani_en.htm.
- [18] MUSSE, JUNG, JACQUES, BRAUN, Using computer vision to simulate the motion of virtual agents
- [19] HELBING, FARKAS e VICSEK, Simulating dynamical features of escape panic
- [20] www.red3d.com/cwr/boids/

Apêndice 2 - Código Fonte

breed [peoples people]

peoples-own [fuzzy-parameter heading-to-center dis speed speeds speeda speedc

speed-near speed-medium speed-far distance-near distance-medium distance-far

distance-from-nearest-neighbor canWalk nearest-neighbor flockmates

heading-difference difference-near difference-medium difference-far

difference-to-center-of-flocking difference-to-center-of-flocking-near difference-to-center-of-flocking-medium difference-to-center-of-flocking-far

nearest-wall distance-to-wall distance-to-wall-near distance-to-wall-medium distance-to-wall-far]

globals [x i j y aux valorFuzzy wall angle angleR angleL]

patches-own []

to setup

;ask patches [set plabel pxcor]

set-default-shape peoples "directional-circle";

set i 0

create-peoples numberOfPeople [configPeople]

end

..... SETUP PEOPLE

```
to configPeople
```

```
  setxy random-xcor random-ycor
```

```
; set heading 90
```

```
  set size 2
```

```
end
```

```
..... GO
```

```
to go
```

```
  if count turtles = 0 [stop]
```

```
  ask peoples
```

```
  [
```

```
    if pcolor = blue [ die]
```

```
    ifelse leave-trace [pd] [pu]
```

```
      set wall patches in-cone avoid-wall-radius-length avoid-wall-radius-angle with
```

```
[pcolor = blue]
```

```
      ifelse any? wall
```

```
        [
```

```
          if wal[
```

```
            avoid-wall
```

```
]
fd 0.6
]
[
flock
]
]
tick
end
```

```
..... FLOCKING
.....
```

```
to flock

set flockmates other turtles in-cone vision-radius-length vision-radius-angle

ifelse any? flockmates

[
set speeds 0.2

if sep [
set flockmates other turtles in-cone separate-radius-length separate-radius-angle

separate

fd speeds

]
```

```
if ali [  
  set flockmates other turtles in-cone align-radius-length align-radius-angle  
  align  
  fd speeds  
]  
if coh  
  [  
    set flockmates other turtles in-cone cohere-radius-length cohere-radius-angle  
    cohere  
    fd speeds  
  ]  
]  
[  
  if sep [fd 0.2]  
  if ali [fd 0.2]  
  if coh [fd 0.2]  
]  
end
```

```
;;; SEPARATE
```

```
to separate
```

```
  if any? flockmates
```

```
  [
```

```
    set heading-difference (subtract-headings heading average-heading-towards-  
flockmates)
```

```
    set aux 180
```

```
    set difference-near precision ((fuzzyficar (0.25 * aux) (0.25 * aux) (0.5 * aux) abs  
heading-difference)) 3
```

```
    set difference-medium precision ((fuzzyficar (0.25 * aux) (0.5 * aux) (0.75 * aux) abs  
heading-difference)) 3
```

```
    set difference-far precision ((fuzzyficar (0.5 * aux)(0.75 * aux)(0.75 * aux) abs heading-  
difference)) 3
```

```
;set speeds difference-near * 0.0 + difference-medium * 0.1 + difference-far * 0.2
```

```
set aux max (list difference-near difference-medium difference-far)
```

```
if (aux = difference-near)
```

```
[set y precision (difference-near * max-turn-separate / 4) 3]
```

```
if (aux = difference-medium)
```

```

[set y precision (difference-medium * max-turn-separate / 2) 3]
    if (aux = difference-far)
[set y precision (difference-far * max-turn-separate) 3]
;show y
ifelse(heading-difference > 0)
[rt y]
[lt y]
]
end

;;; ALIGN
to align ;; turtle procedure
if any? flockmates
[
set heading-difference (subtract-headings average-flockmate-heading heading)
set aux 180

set difference-near precision ((fuzzyficar (0.25 * aux) (0.25 * aux) (0.5 * aux) abs
heading-difference)) 3

set difference-medium precision ((fuzzyficar (0.25 * aux) (0.5 * aux) (0.75 * aux) abs

```

```

heading-difference)) 3

  set difference-far precision ((fuzzyficar (0.5 * aux)(0.75 * aux)(0.75 * aux) abs heading-
difference)) 3

; set speeda difference-near * 0.0 + difference-medium * 0.1 + difference-far * 0.2

  set aux max (list difference-near difference-medium difference-far)

; show aux

  if (aux = difference-near)
    [set y precision (difference-near * max-turn-align / 4) 3]
  if (aux = difference-medium)
    [set y precision (difference-medium * max-turn-align / 2) 3]
    if (aux = difference-far)
      [set y precision (difference-far * max-turn-align) 3]

; show y

; set y precision (difference-near * max-turn-align / 4 + difference-medium * max-turn-
align / 2 + difference-far * max-turn-align) 3

  ifelse(heading-difference > 0)
    [rt y]

```

```
[lt y]
]
end
```

```
to-report average-flockmate-heading ;; turtle procedure
;; We can't just average the heading variables here.
;; For example, the average of 1 and 359 should be 0,
;; not 180. So we have to use trigonometry.
let x-component sum [sin heading] of flockmates
let y-component sum [cos heading] of flockmates
ifelse x-component = 0 and y-component = 0
  [ report heading ]
  [ report atan x-component y-component ]
end
```

```
;;; COHERE
```

```
to cohere ;; turtle procedure
if any? flockmates
[
  set heading-difference (subtract-headings average-heading-towards-flockmates
```

heading)

set aux 180

set difference-near precision ((fuzzyficar (0.25 * aux) (0.25 * aux) (0.5 * aux) abs heading-difference)) 3

set difference-medium precision ((fuzzyficar (0.25 * aux) (0.5 * aux) (0.75 * aux) abs heading-difference)) 3

set difference-far precision ((fuzzyficar (0.5 * aux)(0.75 * aux)(0.75 * aux) abs heading-difference)) 3

; set speedc difference-near * 0.0 + difference-medium * 0.1 + difference-far * 0.2

set aux max (list difference-near difference-medium difference-far)

if (aux = difference-near)

[set y precision (difference-near * max-turn-cohere / 4) 3]

if (aux = difference-medium)

[set y precision (difference-medium * max-turn-cohere / 2) 3]

if (aux = difference-far)

[set y precision (difference-far * max-turn-cohere) 3]

;show y

```
ifelse(heading-difference > 0)
  [rt y]
  [lt y]
]
end
```

```
to-report average-heading-towards-flockmates ;; turtle procedure
```

```
;; "towards myself" gives us the heading from the other turtle
```

```
;; to me, but we want the heading from me to the other turtle,
```

```
;; so we add 180
```

```
if (count flockmates > 0)
```

```
[ let x-component mean [sin (towards myself + 180)] of flockmates
```

```
let y-component mean [cos (towards myself + 180)] of flockmates
```

```
ifelse x-component = 0 and y-component = 0
```

```
  [ report heading ]
```

```
  [ report atan x-component y-component ]]
```

```
report heading
```

```
end
```

```
;;; HELPER PROCEDURES
```

```
to turn-towards [new-heading max-turn] ;; turtle procedure
  turn-at-most (subtract-headings new-heading heading) max-turn
end
```

```
to turn-away [new-heading max-turn] ;; turtle procedure
  turn-at-most (subtract-headings heading new-heading) max-turn
end
```

```
;; turn right by "turn" degrees (or left if "turn" is negative),
;; but never turn more than "max-turn" degrees
```

```
to turn-at-most [turn max-turn] ;; turtle procedure
ifelse turn > 0
  [rt max-turn ]
  [lt max-turn ]
end
```

```
..... FUZZY
```

```
to-report fuzzyficar [esquerda meio direita valorX]
;show esquerda show meio show direita valorX
ifelse (esquerda = meio)
  [ifelse (valorX <= esquerda)
```

```

    [set valorFuzzy 1]
    [ifelse (valorX >= direita)
    [set valorFuzzy 0]
    [set valorFuzzy (valorX - direita) / (meio - direita)]
    ]
]
[ifelse (meio = direita)
    [ifelse (valorX >= direita)
    [set valorFuzzy 1]
    [ifelse (valorX <= esquerda)
    [set valorFuzzy 0]
    [set valorFuzzy (valorX - esquerda) / (meio - esquerda)]
    ]
    ]
    [ifelse( valorX <= esquerda or valorX >= direita )
    [set valorFuzzy 0];
    [ifelse( valorX <= meio )
    [set valorFuzzy (valorX - esquerda) / (meio - esquerda)];
    [set valorFuzzy (valorX - direita) / (meio - direita)];
    ]
    ]
]

```

```

]
report valorFuzzy
end
;;;; SPEED
to calculate-speed
  set flockmates other turtles in-cone (vision-radius-length + size) vision-radius-angle
  ifelse any? flockmates
    [
      set nearest-neighbor min-one-of flockmates [distance myself]
      set dis (precision (distance nearest-neighbor) 3)
      set speed-near precision ((fuzzyficar (0.25 * vision-radius-length) (0.25 * vision-
radius-length) (0.5 * vision-radius-length) dis)) 3
      set speed-medium precision ((fuzzyficar (0.25 * vision-radius-length) (0.5 * vision-
radius-length) (0.75 * vision-radius-length) dis)) 3
      set speed-far precision ((fuzzyficar (0.5 * vision-radius-length) (0.75 * vision-
radius-length) (0.75 * vision-radius-length) dis)) 3
      set speed precision (speed-far * max-speed + speed-medium * max-speed / 2 +
speed-near * 0) 3
    ];else
    [set speed max-speed]
  end
end

```

..... VISION
.....

to show-separate

ask patches in-cone separate-radius-length separate-radius-angle [set pcolor green]

end

to show-align

ask patches in-cone align-radius-length align-radius-angle [set pcolor yellow]

;ask patches in-cone align-radius-length max-turn-align [set pcolor red]

end

to show-cohere

ask patches in-cone cohere-radius-length cohere-radius-angle [set pcolor brown]

;ask patches in-cone (cohere-radius-length - 1) cohere-radius-angle [set pcolor black]

;ask patches in-cone cohere-radius-length max-turn-cohere [set pcolor yellow]

end

to show-vision

ask patches in-cone vision-radius-length vision-radius-angle [set pcolor grey]

end

to show-avoid-wall


```
        ifelse(subtract-headings heading (towards nearest-wall) > 0)
            [rt y]
            [lt y]
        ];else
        [ ]

end
```

```
..... DRAW
```

```
to draw-obstacle
```

```
    draw green
```

```
end
```

```
to draw-fire
```

```
    draw red
```

```
end
```

```
to draw-exit
```

```
    draw yellow
```

```
end
```

```
to draw-wall
```

```
  draw blue
```

```
end
```

```
to draw-black
```

```
  draw black
```

```
end
```

```
to draw [colour]
```

```
  if mouse-down?
```

```
  [
```

```
    set x mouse-xcor
```

```
    set y mouse-ycor
```

```
    ask patches [
```

```
      if ((abs (pxcor - x)) < size-of-pen) and ((abs (pycor - y)) < size-of-pen)
```

```
        [ set pcolor colour ]
```

```
    ]
```

```
  ]
```

```
end
```

