



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Joelmir Ramos da Costa

**Deteccão de Anomalias em Vídeos com Multidão
Utilizando Colônia de Bactérias Artificiais**

Rio de Janeiro
2017

Joelmir Ramos da Costa

**Detecção de Anomalias em Vídeos com Multidão
Utilizando Colônia de Bactérias Artificiais**



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Orientadora: Prof.^a Dr.^a Nadia Nedjah

Orientadora: Prof.^a Dr.^a Luiza de Macedo Mourelle

Rio de Janeiro
2017

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

C837 Costa, Joelmir Ramos da
Detecção de Eventos Anormais em Vídeos com Multi-
dão Utilizando Colônia de Bactérias Artificiais/Joelmir
Ramos da Costa:- 2017.

121 122f.

Orientadora: Nadia Nedjah.

Orientadora: Luiza de Macedo Mourelle.

Dissertação (Mestrado) – Universidade do Estado do
Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia Eletrônica. 2. Fluxo óptico - Teses. 3.
Inteligência coletiva - Teses. 4. Redes neurais (Compu-
tação) - Teses. I. Nedjah, Nadia. II. Mourelle, Luiza de
Macedo. III. Universidade do Estado do Rio de Janeiro.
IV. Título.

CDU 004.032.26

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta
dissertação, desde que citada a fonte.

Assinatura

Data

Joelmir Ramos da Costa

Detecção de Anomalias em Vídeos com Multidão Utilizando Colônia de Bactérias Artificiais

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Aprovado em: 1 de agosto de 2017

Banca Examinadora:

Prof.^a Dr.^a Nadia Nedjah (Orientadora)
Faculdade de Engenharia, UERJ

Prof.^a Dr.^a Luiza de Macedo Mourelle (Orientadora)
Faculdade de Engenharia, UERJ

Prof. Dr. Moacir Antonelli Ponti
Instituto de Ciências Matemáticas e de Computação, USP

Profa. Dra. Flavia C. Bernardini
Departamento de Computação, UFF

Rio de Janeiro
2017

DEDICATÓRIA

Dedico este trabalho à minha mãe, que acreditou em meus esforços e me nutriu de amor a cada passo da minha jornada.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela vida e pelas oportunidades que tenho. Às professoras Nadia Nedjah e Luiza de Macedo Mourelle pela orientação, pelos ensinamentos e pelos incentivos dados ao longo do desenvolvimento deste trabalho.

Agradeço a minha mãe pelo incentivo nos momentos de maior dificuldade e ao meu pai pelo apoio durante a minha caminhada. Agradeço a Marília Beatriz pelas instruções e direcionamento.

Agradeço ao meu grande irmão Daniel Ramos que me ajudou a seguir nos momentos em que eu quase não tinha mais forças para continuar.

Agradeço aos meus amigos de longa data Diego Jorge, Adriano Barros e Leonardo Soares, por todo apoio ao longo dos anos.

Agradeço aos meus colegas de mestrado Luneque, Alejandra, Alexandre, Igor, Pedro e Tarso, pela troca de conhecimentos, pelo companheirismo e pelos momentos de descontração.

Agradeço a Dayane dos Anjos pela inspiração que me fez seguir em frente e pela paciência durante os períodos de maior empenho e dedicação ao trabalho.

Agradeço à CAPES e ao PEL-UERJ pelos recursos investidos nesta pesquisa.

“O homem erra porque ainda não sabe”
Platão

RESUMO

COSTA, Joelmir Ramos da. *Detecção de Anomalias em Vídeos com Multidão Utilizando Colônia de Bactérias Artificiais*. 2017. 121f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2017.

Em um mundo cada vez mais populoso, em que grandes quantidades de pessoas se aglomeram em espaços públicos diariamente, a compreensão e previsão do comportamento humano em multidões são essenciais para manter a segurança dos indivíduos. Atualmente, com o aumento constante do número de câmeras de vigilância espalhadas por todo o mundo, a análise da atividade humana em larga escala tornou-se possível. No entanto, o volume de dados a ser analisado e classificado, além do custo computacional envolvido no processo, tornam o desenvolvimento de sistemas de detecção de eventos precisos em tempo real um grande desafio. Nesta dissertação, desenvolve-se um sistema de detecção de eventos em vídeos com multidão, que possibilita aplicações em tempo real. O sistema proposto neste trabalho avalia o tempo de processamento de três métodos distintos de extração de movimento entre *frames* e de um algoritmo de otimização inspirado em colônias de bactérias, que recobre com bactérias artificiais as regiões de interesse das camadas contendo movimento. Por último, utiliza-se redes neurais de Kohonen para classificar os padrões de comportamento das colônias que emergem durante a otimização. Com base no método proposto, foram avaliados dois consagrados conjuntos de dados na análise de eventos em vídeos de curta duração, contendo multidões de média e alta densidade: UMN e PETS 2009. Avaliou-se também um vídeo de vigilância de longa duração com imagens do tráfego de veículos em uma avenida. Todas as simulações foram realizadas no MATLAB[®]. O desempenho dos métodos de extração de movimento foi avaliado de acordo com o tempo de processamento por par de *frames*. O desempenho do algoritmo de Colônia de Bactérias é mensurado pelo tempo de processamento e pela quantidade de bactérias iniciais na camada de movimento, e a qualidade do classificador é comparada com o estado da arte de outros sistemas de detecção de eventos em vídeos, através da área sob a curva ROC, tendo apresentado resultados semelhantes, porém com baixo custo computacional e possibilidades de aplicação em tempo real. Os resultados das simulações e dos experimentos mostram a eficácia e eficiência do sistema proposto.

Palavras-chave: Fluxo Óptico; Modelo de Força Social; Diferença Absoluta entre *Frames*; Inteligência Coletiva; Redes Neurais de Kohonen; Otimização por Colônia de Bactérias Artificiais; Detecção de Anomalias em Vídeos.

ABSTRACT

COSTA, Joelmir Ramos da. *Detection of anomalies in Crowd Videos Using Artificial Bacteria Colony*. 2017. 121f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2017.

In an increasingly populated world wherein large crowd are expected in public spaces on a daily basis, understanding and predicting human behavior within crowds is essential for maintaining the safety of individuals. Nowadays, with the steady increase in the number of surveillance cameras, distributed throughout the world, the analysis of human activity on a large scale has become possible. However, the volume of data to be analyzed and classified, in addition to the computational cost involved in the process, make the development of accurate real-time event detection systems a major challenge. In this dissertation, an event detection in crowd video system is developed that allows real-time applications. The proposed system in this work evaluates the performance of 3 methods for motion extraction between *frames*, develops an optimization algorithm inspired by bacterial colonies, aiming at quickly covering with artificial bacteria the regions of interest of the layers containing movement, and finally, it uses Kohonen neural networks to classify the behavioral patterns of bacteria colonies that emerge during the optimization. Based on the proposed method, we evaluated two video datasets which are commonly used about medium and high density of crowds: UMN and PETS 2009. We also evaluated a long surveillance video about of car traffic on a high speed avenue. All the simulations were performed on MATLAB[®]. The performance of the motion extraction methods is evaluated according to the processing time per pair of frames. The performance of the algorithm CBA is measured by the processing time and the amount of initial bacteria and the classifier is compared to that of other video event detection state-of-the-art algorithms by the area under the ROC curve, presenting similar results but with lower computational cost and real-time application possibilities. The results of the experiments demonstrate the efficacy and efficiency of the proposed system.

Keywords: Optical Flow; Social Force Model; Absolute Difference Between Frames; Collective Intelligence; Optimization by Artificial Bacteria Colony Algorithm; Kohonen's Neural Network; Abnormalities Detection in Videos.

LISTA DE FIGURAS

1	Ilustração do fluxo óptico (FO)	24
2	Representação do problema de abertura para 3 diferentes padrões de movimento	25
3	Fluxo óptico	27
4	Reprodução bacteriana	44
5	Ilustração da quimiotaxia.	45
6	Movimento de nado e giro da bactéria <i>Escherichia coli</i>	46
7	Quimiotaxia na função de custo.	47
8	Comportamento de convergência das bactérias virtuais nas curvas de nível da função <i>Sphere</i> (PASSINO, 2005)..	52
9	Estrutura neural básica da rede competitiva	56
10	Ajuste de pesos do neurônio vencedor	57
11	Representação dos vetores de peso	58
12	Distribuição tridimensional dos vetores de pesos em relação ao conjunto de amostras (SILVA; SPATTI; FLAUZINO, 2010).	58
13	Mapa topológico unidimensional da RNK	61
14	Mapa topológico bidimensional da RNK	61
15	Ilustração do decaimento Gaussiano para o ajuste dos pesos dos neurônios vizinhos ao vencedor.	62
16	Exemplo de um mapa de contexto para a topologias bidimensional	64
17	Diagrama de blocos do sistema proposto	67
18	Em (a) é mostrado um par de <i>frames</i> consecutivos, (b) mostra a DAF do par de <i>frames</i> , (c) mostra o FO dos <i>frames</i> e (d) mostra o FF gerado pelo MFS a partir do FO mostrado em (c).	71
19	Em (a) é mostrado uma colônia de bactérias reais e em (b) as bactérias reais demarcando um ambiente nutricional na forma de mão. Em (c) mostra-se uma camada de movimento extraída de 2 <i>frames</i> consecutivos e em (d), mostra-se a distribuição das bactérias artificiais sobre essa camada.	72
20	Ilustração da otimização de uma camada de movimento.	77
21	Função de custo corresponde à camada de movimento da Figura 20(b).	78
22	Função de custo da Figura 21 recoberta com as bactérias artificiais após a otimização. Essa função equivale à camada otimizada mostrada na Figura 20(d).	78
23	Evolução da colônia de bactérias ao longo de 5 épocas da camada de movimento apresentada na Figura 20(b). O pequeno triângulo verde em (e) representa a posição do centroide.	79
24	Janela de simulação da ferramenta SOM	81

LISTA DE FIGURAS

25	Topologias	83
26	Posição dos neurônios antes e depois do treinamento	83
27	<i>Frames</i> representando os vídeos do <i>dataset</i> PETS2009.	88
28	Tempo médio de processamento para geração de uma camada de movimento a partir de dois <i>frames</i>	89
29	Tempo médio de otimização por camada de movimento obtido pelo FO.	90
30	Tempo médio de otimização por camada de movimento obtido pelo FF.	91
31	Tempo médio de otimização por camada de movimento obtido pela DAF.	91
32	Registro das populações, estoques e centroides das colônias de bactérias obtidas ao longo da otimização do vídeo 1 <i>H</i> L, para 250 bactérias iniciais.	92
33	Registro das populações, estoques e centroides das colônias de bactérias obtidas ao longo da otimização do vídeo 1 <i>H</i> L, para 500 bactérias iniciais.	92
34	Distribuição dos pesos da RNK após o treinamento.	93
35	Comparação entre as classes esperadas, que possuem um evento considerado anormal, e as classes obtidas pelo método proposto para todos os frames do <i>dataset</i> PETS2009.	95
36	Curva ROC das detecções realizadas em todos os vídeos do <i>dataset</i> PETS2009 utilizando as 3 diferentes técnicas de extração de movimento.	95
37	<i>Frames</i> representando os vídeos do <i>dataset</i> UMN.	96
38	Tempo médio de processamento para geração de uma camada de movimento a partir de dois <i>frames</i> consecutivos.	97
39	Tempo médio de otimização por camada de movimento obtido pelo FO.	98
40	Tempo médio de otimização por camada de movimento obtido pelo FF.	98
41	Tempo médio de otimização por camada de movimento obtido pela DAF.	98
42	Registro das populações, estoques e centroides das colônias de bactérias obtidas ao longo da otimização do vídeo 1, para camadas de movimento geradas pelo FO, FF e DAF.	99
43	Distribuição dos neurônios da RNK para os registros de otimização do vídeo 1, para 500 bactérias iniciais e camadas de movimento extraídas pela DAF. Para melhor visualização da imagem, os valores de estoque foram divididos por 5 em (b).	100
44	Comparação entre as classes desejadas, que possuem um evento considerado anormal, e as classes obtidas pelo método proposto para todos os frames do <i>dataset</i> UMN. Esta figura refere-se apenas ao resultado obtido utilizando a DAF para a extração de movimento e 500 bactérias iniciais na otimização.	101
45	Curva ROC das detecções realizadas em todos os vídeos do <i>dataset</i> UMN utilizando as 3 diferentes técnicas de extração de movimento	101
46	<i>Frames</i> representando os dois eventos distintos considerados normais.	103
47	Comportamento das populações de bactérias, com destaque para os padrões que emergem da otimização.	104
48	Tempo médio para a extração de movimento e otimização das camadas, para cada parte do vídeo.	104
49	Distribuição dos neurônios da RNK para os registros de otimização de todas as partes do vídeo para 500 bactérias iniciais e camadas de movimento extraídas pela DAF. (a) revela a distribuição bidimensional da população x centroide e (b) a distribuição da população x estoque.	105
50	Visualização dos dados em 3 dimensões	105

LISTA DE FIGURAS

51	Detecção após o treinamento da rede.	106
52	<i>Frames</i> correspondentes aos eventos destacados na Figura 51.	106
53	Curva ROC para a classificação de todos os <i>frames</i> do vídeo.	106

LISTA DE TABELAS

1	BFO <i>vs.</i> CBA	77
2	<i>Speedup</i> para o <i>dataset</i> PETS2009	90
3	Número de neurônios, tempo médio e desvio padrão do treinamento, e tempo de classificação após o treino para cada vídeo do <i>dataset</i> PETS2009	94
4	Área sob a curva ROC do método proposto <i>vs.</i> outros métodos, para o <i>dataset</i> PETS 2009	95
5	<i>Speedup</i> no pré-processamento dos vídeos do <i>dataset</i> UMN	97
6	Número de neurônios, tempo médio de treinamento e desvio padrão, e tempo médio de classificação para uma amostra após o treino, para cada vídeo do <i>dataset</i> UMN.	100
7	Área sob a curva ROC do método proposto <i>vs.</i> outros métodos para o <i>dataset</i> UMN.	102
8	Especificações do vídeo	102

LISTA DE ALGORITMOS

1	Algoritmo BFO	50
2	Algoritmo Competitivo – Fase de Treinamento	59
3	Algoritmo Competitivo – Fase de Operação	60
4	Algoritmo de treinamento da RNK	63
5	Algoritmo de operação da RNK	64
6	Algoritmo de colônia de bactérias artificiais (CBA)	76

LISTA DE SIGLAS

ABC	<i>Artificial Bee Colony</i>
ACO	<i>Ant Colony Optimization</i>
AMC	<i>Adjacency-Matrix Based Clustering</i>
AUC	<i>Area Under ROC Curve</i>
BCA	<i>Bacterial Chemotaxis Algorithm</i>
BFO	<i>Bacterial Foraging Optimization</i>
CBA	<i>Colônia de Bactérias Artificiais</i>
CR	<i>Condição de Reprodução</i>
CS	<i>Condição de Sobrevivência</i>
FF	<i>Fluxo de Forças</i>
FO	<i>Fluxo Óptico</i>
FPGA	<i>Field Programmable Gate Array</i>
GMM	<i>Approximate Median Method</i>
GMM	<i>Gaussian Mixture Model</i>
GPU	<i>Graphics Processing Unit</i>
HGO	<i>Histograms of Oriented Gradients</i>
HOS	<i>Histograms of Oriented Swarms</i>
HOSA	<i>Histograms of Oriented Swarm Accelerations</i>
MFS	<i>Modelo de Força Social</i>
OPF	<i>Optimal Power Flow</i>
PID	<i>Proportional Integral Derivative</i>
PSO	<i>Particle Swarm Optimization</i>
RFO	<i>Restrição do Fluxo Óptico</i>
RGB	<i>Red Gren Blue</i>

RNK	<i>Rede Neural de Kohonen</i>
ROC	<i>Receiver Operating Characteristic</i>
SCOF	<i>Statistical Characteristic of the Optical Flow</i>
SOM	<i>Self Organizing Maps</i>
SVM	<i>Support Vector Machine</i>
SVR	<i>Support Vector Regression</i>

SUMÁRIO

INTRODUÇÃO	17
1 EXTRAÇÃO DE MOVIMENTO	23
1.1 Fluxo Óptico	23
1.1.1 <u>Método de Lucas e Kanade</u>	26
1.1.2 <u>Método de Horn e Schunck</u>	27
1.1.3 <u>Método de Augereau</u>	28
1.1.4 <u>Outros Métodos</u>	29
1.2 Modelo de Força Social	30
1.3 Diferença Absoluta entre <i>Frames</i>	31
1.4 Considerações Finais	33
2 TRABALHOS RELACIONADOS	34
2.1 Análise de Cenas por FO	34
2.2 Análise de Cenas por MFS	36
2.2.1 <u>Métodos Determinísticos</u>	37
2.2.2 <u>Métodos Estocásticos</u>	39
2.3 Considerações Finais	40
3 OTIMIZAÇÃO BASEADA NO FORRAGEAMENTO DE BAC- TÉRIAS	41
3.1 Inteligência de Enxame	41
3.2 Bactérias Reais	43
3.3 Quimiotaxia	44
3.4 Algoritmo de Otimização por Forrageamento	47
3.5 Aplicações	52
3.6 Considerações Finais	54
4 REDES NEURAIIS DE KOHONEN	55
4.1 Aprendizado Competitivo	55
4.2 Redes Neurais de Kohonen	60
4.3 Considerações Finais	65
5 ASPECTOS DE IMPLEMENTAÇÃO	66
5.1 Arquitetura do Sistema	66
5.2 Extração de Movimento	68
5.2.1 <u>Diferença Absoluta entre <i>Frames</i></u>	68
5.2.2 <u>Fluxo Óptico</u>	68
5.2.3 <u>Fluxo de Forças</u>	69
5.3 Colônia de Bactérias Artificiais	71

SUMÁRIO

5.4	Redes Neural de Kohonen.....	80
5.5	Considerações Finais	84
6	EXPERIMENTOS E RESULTADOS	85
6.1	Métricas Utilizadas.....	85
6.2	Configurações do Algoritmo	87
6.3	<i>Dataset</i> PETS 2009	88
6.4	<i>Dataset</i> UMN.....	95
6.5	Video de Longa Duração.....	102
6.6	Considerações Finais	107
7	CONCLUSÕES E TRABALHOS FUTUROS	108
7.1	Conclusões	108
7.2	Trabalhos Futuros.....	110
	REFERÊNCIAS	112

INTRODUÇÃO

UMA das tarefas mais desafiadoras na área de computação visual é a análise da atividade humana em vídeos com multidão. O comportamento que emerge da auto-organização entre as pessoas apresenta-se como um obstáculo para o reconhecimento de eventos bem definidos em cenas com tumulto (HELBING et al., 2001).

O fenômeno frequente de grandes aglomeração de pessoas em espaços públicos e o crescente número de câmeras de segurança ativas em todo o planeta, estimulam a busca por sistemas automatizados de detecções de eventos potencialmente perigosos. A detecção automatizada de tais eventos é importante porque a condição humana não permite que poucos indivíduos sejam capazes de realizar esta tarefa manualmente em larga escala. A detecção de eventos incomuns nos vídeos confronta-se com a dificuldade de apreender o padrão de comportamento das pessoas, compensar a baixa resolução das câmeras de vigilância que geram demasiado ruído e, principalmente, fornecer o custo computacional exigido pelo sistema (ZITOUNI et al., 2016).

O desenvolvimento de sistemas de detecção de eventos em vídeos, especialmente as anomalias, é de grande interesse das áreas de computação visual voltadas para a vigilância e monitoramento. Do mesmo modo, a compreensão do fenômeno de multidão é também do interesse de diversos campos de pesquisa, e útil em várias aplicações, tais como:

- Gestão de multidão: a análise do comportamento de multidão pode ser utilizada para o desenvolvimento de estratégias de gestão de público, especialmente nos eventos mais populares, tais como jogos esportivos, grandes shows, manifestações públicas, etc. Estas estratégias ajudam a evitar desastres e garantem a segurança pública.
- Concepção de espaços públicos: a análise do comportamento de multidão pode fornecer orientações para a concepção de espaços públicos, como por exemplo, determinar o *layout* de centros comerciais mais convenientes para clientes ou para facilitar os acessos a locais onde concentram-se muitas pessoas.

- Ambientes virtuais: os modelos matemáticos que representam uma aproximação do comportamento de multidão podem ser empregados em ambientes virtuais para melhorar a simulação de multidões e, conseqüentemente, enriquecer a compreensão do comportamento humano.
- Vigilância visual: a análise do comportamento de multidões pode ser usada para detecção automática de anormalidades. Além disso, a capacidade de rastrear indivíduos em uma multidão poderia ajudar, por exemplo, as autoridades a identificarem suspeitos.

Gestão de multidões e desenvolvimento de espaços públicos são alvos de pesquisas de sociólogos, psicólogos e engenheiros civis. As abordagens favorecidas pela psicologia e sociologia são baseadas na observação e análise do comportamento humano. Sociólogos, por exemplo, estudam os padrões de comportamento das multidões como fenômenos sociais. Métodos computacionais, por outro lado, tais como os empregados em computação gráfica e métodos de análise focados na extração de características quantitativas e detecção de eventos em multidões, sintetizam o fenômeno matematicamente, com modelos estatísticos (ZHAN et al., 2008).

A análise de movimento e comportamentos de multidão constitui uma tarefa desafiadora para os métodos tradicionais da computação visual, principalmente os comportamentos associados à oclusão de indivíduos e variação aleatória na densidade de pessoas. O desafio consiste principalmente em apreender a natureza estocástica complexa dos movimentos nos vídeos e eliminar os distúrbios provenientes dos ruídos. Além disso, o custo computacional é outro fator que torna difícil o desenvolvimento de um método que solucione esses problemas. Em muitas situações práticas, é crucial analisar as cenas com multidão em tempo real, ou no mínimo, tão rápido quanto possível, considerando o fato de que este tipo de sistema deve agir rapidamente se algo parecer não estar dentro dos parâmetros de normalidade (CAO et al., 2009).

O objetivo principal da análise de vídeos com multidões é realizar a modelagem da dinâmica de massa de pessoas para detectar ou mesmo prever possíveis comportamentos anormais nas cenas. Para os cenários particulares de vigilância, essa tarefa é de grande importância, visto que a rápida detecção e predição de anormalidades podem reduzir possíveis conseqüências danosas de um evento catastrófico ou alertar o operador humano para inspecionar mais cuidadosamente a cena do evento (ZITOUNI et al., 2016).

Existem diversas técnicas que possibilitam a análise de vídeos com multidão. A literatura, no entanto, revela que o Fluxo Óptico (FO) encontra-se entre as melhores técnicas para extrair o movimento de cenas com estabilidade e baixo ruído (TZIAKOS; CAVALLARO; XU, 2010; CHAARAOU; CLIMENT-PÉREZ; FLÓREZ-REVUELTA, 2012; KAJO; MALIK; KAMEL, 2015). O FO produz um campo de vetores contendo as magnitudes das variações de brilho dos *pixels* entre dois *frames* consecutivos. Isso permite obter camadas contendo as variações de movimento em sequências de imagens. Essa técnica, no entanto, ainda é sensível às variações bruscas de luminosidade do ambiente, além de demandar significativo custo computacional.

Na tentativa de proporcionar melhorias na apreensão do movimento entre *frames*, o Modelo de Força Social (MFS) proposto em (HELBING; MOLNAR, 1995) simula com sucesso o fenômeno de auto-organização do comportamento de pedestres (YAN; LI; MUCHAO, 2012). No MFS, a interação entre pedestres é mensurada através de forças sociais. Em (MEHRAN; OYAMA; SHAH, 2009), os campos de vetores resultantes das forças sociais, ou fluxo de forças (FF), são obtidos a partir dos campos de vetores do FO. Assim, o MFS captura a dinâmica das forças de interação da multidão com base no FO, porém gera inevitavelmente um incremento na demanda por processamento computacional.

A utilização de algoritmos mais simples, que realizam a separação entre o plano de fundo (*background*) e o primeiro plano (*foreground*), permitem extrair simplificada o movimento entre *frames* e promovem uma significativa redução no esforço computacional (SEESE et al., 2016; KHILAR; CHITRAKALA, 2014). Esses algoritmos destacam as regiões de interesse. A extração de *foreground* é um método comum de segmentação e consiste basicamente em calcular a diferença *pixel a pixel* entre o *frame* atual e o seguinte. Se a diferença for menor que um limiar pré-definido, esse *pixel* pertence ao fundo, caso contrário, pertence ao objeto em movimento. Assim, essa técnica extremamente simples captura o movimento dos *frames*, no entanto o uso do limiar estático introduz, conseqüentemente, elevado ruído. Nesta dissertação, utilizou-se a diferença absoluta entre *frames* (DAF) (BILAL; KHAN, 2015) para destacar as regiões com movimento. Comparada às técnicas que utilizam FO e FF, a DAF produz uma extração de movimento similar, porém com grande vulnerabilidade a qualquer tipo de influência ruidosa.

Nesta dissertação, a análise das cenas do vídeo inicia-se a partir das camadas de movimento geradas por 3 técnicas de extração: FO, FF e DAF. Recentemente, a

busca otimizada por regiões de interesse com algoritmos inspirados em inteligência de enxame tem se mostrado promissora (THIDA; ENG; REMAGNINO, 2013; KAL TSA et al., 2015; SEESE et al., 2016). Na inteligência de enxame, os algoritmos desenvolvidos com base no comportamento coletivo dos seres vivos realizam buscas leves e rápidas por valores ótimos nas funções de custo. O algoritmo de otimização por forrageamento de bactérias (BFO) é uma metaheurística inspirada no comportamento de forrageamento da *Escherichia coli* e apresenta vantagens computacionais tanto na otimização de funções multi-dimensionais (PASSINO, 2005, 2012; JADHAV; DEVI; PAT TNAIK, 2012) quanto na otimização de imagens (SAADI et al., 2010) e vídeos (NGUYEN; BHANU, 2012).

Neste trabalho, uma metaheurística inspirada em colônias de bactérias reais é utilizada para recobrir as regiões de interesse nas camadas de movimento. O algoritmo de colônias de bactérias artificiais (CBA) desenvolvido trata-se também de uma modificação e melhoria no algoritmo BFO (PASSINO, 2005). No CBA adicionou-se significativa ênfase à seleção natural e a evolução ao longo das épocas, visando a simplificação e acurácia do algoritmo para a implementação no sistema proposto e possíveis utilizações futuras em aplicações de vigilância em tempo real.

Os padrões de comportamento das bactérias reflete a essência dos padrões de comportamento humano na cena. A imprevisibilidade e a emergência caótica desses padrões inviabilizam o uso tanto de classificadores cujo treinamento é supervisionado quando dos que requerem demasiado processamento na fase de treinamento e classificação. Assim, para atender as especificações do problema e compor o sistema em harmonia com os princípios adaptativos das CBA, são utilizadas nesta dissertação Redes Neurais de Kohonen (RNK), também chamados de mapas auto-organizáveis de Kohonen (SOM) (KOHONEN, 1982, 2012).

As RNK fazem parte de um grupo de redes neurais competitivas, que utilizam estratégias de competição natural para efetuar os ajustes de seus pesos na fase de aprendizado (NURMAINI et al., 2016). São capazes de preservar as relações de vizinhança dos dados de entrada e utilizam treinamento não-supervisionado para encontrar similaridades com base apenas nesses padrões de entrada. O principal objetivo das RNK é agrupar os dados de entrada que são semelhantes entre si formando grupos. As RNK demandam baixo poder computacional na etapa de treinamento devido a sua simplicidade (MARANA et al., 1998).

Nesta dissertação, os conjuntos de treinamento da RNK são os registos comportamentais das populações das bactérias durante a otimização das camadas de movimento. Uma colônia de bactéria corresponde à otimização de uma camada de movimento. Uma camada de movimento representa uma função de custo para o algoritmo CBA. Vetores contendo a população final de cada colônia, seu estoque de alimento acumulado (avaliações da função de custo) e o centroide da posição de todas as bactérias espalhadas pelas regiões de interesse, formam o conjunto de treinamento. Tais registros, em essência, refletem o espalhamento, a quantidade e o núcleo do movimento entre *frames*, respectivamente.

Após o treinamento, o conjunto de dados é agrupado pela RNK segundo as similaridades que apresentam. Assim, são rotulados com base no julgamento humano, que reconhece o evento no vídeo e em seguida realiza a correlação entre o evento e a classe apresentada pela RNK, e também com base nas indicações de trabalhos relacionados (MEHRAN; OYAMA; SHAH, 2009; WU; MOORE; SHAH, 2010; CONG; YUAN; LIU, 2011; LIU; LI; JIA, 2014). O evento anormal seria então, em primeira instância, uma concepção humana a ser identificada na classificação da RNK.

A contribuição deste trabalho está no desenvolvimento de um sistema de detecção de anomalias em vídeos com multidão, que possibilita detecções em tempo real, utilizando a DAF para extrair as camadas de movimento e o algoritmo CBA com 500 bactérias na população inicial, na etapa de otimização destas camadas. A RNK realiza, por último, o agrupamento e a detecção das classes associadas aos eventos anormais. Os primeiros resultados obtidos durante este trabalho foram publicados em (COSTA; NEDJAH; MOURELLE, 2017a) e aceitos para publicação em (COSTA; NEDJAH; MOURELLE, 2017b). O restante desta dissertação está organizado em sete capítulos, cujos conteúdos são brevemente descritos a seguir.

Inicialmente, na apresentação do Capítulo 1, são apresentados alguns métodos importantes que realizam a extração do movimento existente entre *frames*. No Capítulo 2 apresenta-se uma diversidade de trabalhos relacionados ao problema de detecção de eventos em vídeos com multidão. Nesse Capítulo especificamente, os trabalhos relacionados discorrem sobre a utilização do FO em detecções de eventos, seguido de trabalhos que se utilizam do Modelo de Força Social para detectar eventos baseados no FF e, por último, mas não menos importante, são apresentados os trabalhos desenvolvidos tendo como base os métodos estocásticos de detecção.

O Capítulo 3 descreve as técnicas de otimização por inteligência de enxame, com forte ênfase na descrição do algoritmo BFO, que serviu de inspiração para o desenvolvimento do algoritmo CBA. O Capítulo 4 expõe em detalhes a RNK e seu método de treinamento competitivo.

No Capítulo 5, o sistema proposto é descrito em detalhes juntamente com os aspectos relevantes de sua implementação. O Capítulo 6 apresenta a metodologia dos experimentos e os resultados obtidos. E finalmente, no Capítulo 7, apresentam-se as conclusões acerca do método proposto e dos resultados obtidos, bem como algumas perspectivas e possibilidades de trabalhos futuros.

Capítulo 1

EXTRAÇÃO DE MOVIMENTO

ESTIMAR o movimento em uma sequência de *frames* é fundamental para o reconhecimento de eventos em vídeos com multidões. Este capítulo apresenta uma visão geral das técnicas usadas para extrair movimentos dos vídeos. O fluxo óptico (FO), o fluxo de forças (FF), obtido a partir do FO pelo Modelo de Força Social (MFS), e a diferença absoluta entre *frames* (DAF), são utilizadas neste trabalho.

1.1 Fluxo Óptico

Movimento é uma poderosa característica em sequências de imagens, revelando a dinâmica da cena pela relação espacial das características da imagem de acordo com a mudança temporal. A tarefa de análise de movimento é um desafio e um problema fundamental em visão computacional (SPIES; SCHARR, 2001).

A segmentação de movimento para o reconhecimento de eventos em multidões é um tópico importante na área de computação visual. A análise de vídeos tem como objetivo identificar o padrão de comportamento de indivíduos em ambientes distintos. A estimativa da velocidade e direção de grupos na multidão, ou da multidão como um todo, é uma difícil tarefa (CANDAMO et al., 2010). Para lidar com esse desafio, métodos como subtração de fundo (WU; ERDAN; CHUN, 2001), diferença temporal (BARRON; FLEET; BEAUCHEMIN, 1994), correspondência baseada em blocos (PLA et al., 2005) e fluxo óptico têm sido amplamente utilizados (ZITOUNI et al., 2016). A literatura, no entanto, aponta o fluxo óptico como estando entre os métodos mais eficientes para extração de movimento (KAJO; MALIK; KAMEL, 2015).

O fluxo óptico, ou campo de deslocamento, é a distribuição das velocidades aparentes de cada *pixel*, entre duas imagens consecutivas, sob a hipótese de conservação da

luminância. Trata-se de um campo de vetores que contém a taxa de variação de brilho de cada *pixel*, como ilustrado na Figura 1. O fluxo óptico capta tanto o movimento dos objetos da cena quanto o movimento da câmera, e fornece vetores que representam a velocidade e a direção dos *pixels* (HORN; SCHUNCK, 1981).

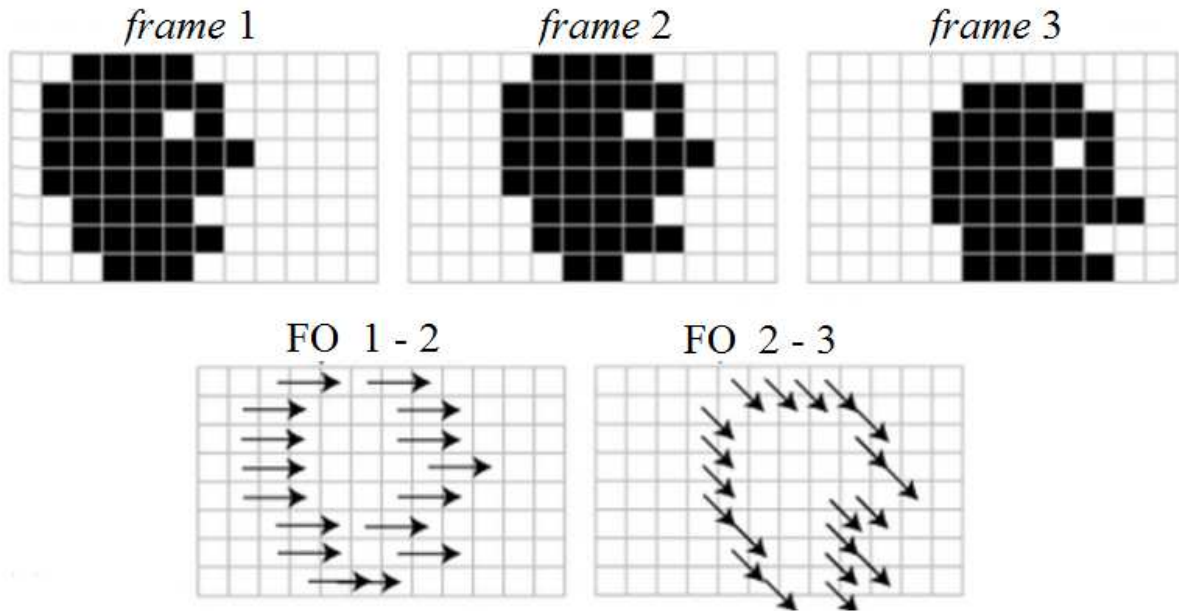


Figura 1: Ilustração do fluxo óptico (FO)

Uma sequência de *frames* pode ser representada pela função de luminância apresentada na Equação 1, em que a luminância é a razão da intensidade do fluxo luminoso emitido por uma superfície em uma dada direção pela área dessa superfície projetada ortogonalmente sobre um plano perpendicular àquela direção.

$$I = I(x_1, x_2, t), \quad (1)$$

onde x_1 e x_2 correspondem as linhas e colunas da imagem, respectivamente, e t representa a posição do *frame* na sequência. A hipótese de conservação da luminância supõe que a luminância de um ponto físico da sequência de imagens não varia durante um curto intervalo de tempo, conforme representado na Equação 2:

$$I(x_1, x_2, t) = I(x_1 + \Delta x_1, x_2 + \Delta x_2, t + \Delta t). \quad (2)$$

onde Δx_1 e Δx_2 representam um pequeno incremento de espaço no eixos e Δt representa o intervalo entre dois *frame* consecutivos. Levando em consideração a hipótese supracitada, tem-se o problema de restrição do fluxo óptico (RFO), representado na Equação 3. A

RFO não possui resolução direta porque o número de equações é inferior ao número de variáveis.

$$\nabla I \cdot \vec{v} + I_t = 0, \quad (3)$$

onde

$$\vec{v} = (v_{x_1}, v_{x_2}, \Delta t). \quad (4)$$

Existem dois grandes desafios para extrair movimentos de vídeos: o primeiro surge do fato de que nem todo movimento na cena está associado aos objetos. Esse problema ocorre devido a alterações no brilho da imagem por causa da baixa resolução do vídeo. Nestes casos, as variações na iluminação são responsáveis por introduzir grandes quantidades de ruído no fluxo óptico. O segundo desafio vem do chamado problema de abertura, ilustrado na Figura 2, em que o movimento circular do objeto em torno do próprio eixo produz um efeito visual que não pode ser capturado pelo fluxo óptico. Para contornar esse problema, o movimento aparente de um ponto é calculado em uma vizinhança pré-definida. Deste modo, é possível calcular a componente do movimento na direção do gradiente, isto é, perpendicular ao contorno (KAJO; MALIK; KAMEL, 2015).

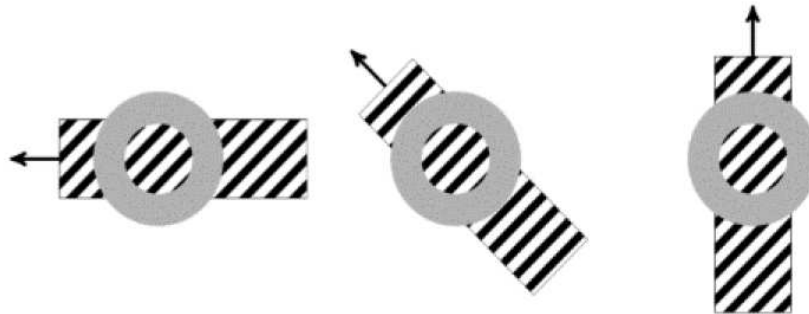


Figura 2: Representação do problema de abertura para 3 diferentes padrões de movimento

Os algoritmos de fluxo óptico são agrupados em duas classes baseando-se no tipo de regularização, que são os métodos baseados em características e os métodos variacionais. Os métodos baseados em características calculam o fluxo óptico para cada *pixel* e sua vizinhança mais próxima, desvinculada de outros grupos de *pixels*. Métodos variacionais levam em consideração o fluxo óptico de *pixels* vizinhos e necessariamente implicam em suposições de suavidade no campo de fluxo (RASHWAN; PUIG; GARCIA, 2012). Numerosos métodos de fluxo óptico foram propostos na literatura. Nesta seção, alguns dos métodos mais populares são discutidos.

Os métodos variacionais para a computação do fluxo óptico podem ser divididos em quatro categorias: métodos diferenciais, métodos por correlação, métodos baseados em energia e métodos baseados em fase. Os principais métodos diferenciais para a computação do fluxo óptico abrangem o método de Lucas e Kanade (LUCAS; KANADE et al., 1981), o método de Horn e Schunck (HORN; SCHUNCK, 1981) e o método de Augereau (AUGEREAU; TREMBLAIS; FERNANDEZ-MALOIGNE, 2005).

1.1.1 Método de Lucas e Kanade

O método de Lucas e Kanade (LUCAS; KANADE et al., 1981; AGARWAL; GUPTA; SINGH, 2016) é uma estimativa de médias ponderadas para o sistema de restrições do fluxo óptico na vizinhança do ponto (x_1, x_2, t) de uma determinada imagem. Este método é amplamente utilizado por ser numericamente estável e resolver a RFO tomando como ponto de partida as médias quadradas de uma vizinhança determinada em uma janela $W(x)$ centrada no ponto (x_1, x_2, t) conforme apresentado na Equação 5:

$$\int_{W(x)} (\nabla_{xt} I \cdot \vec{v})^2 dx' = \int_{W(x)} \vec{v}^T (\nabla_{xt} I) (\nabla_{xt} I)^T \vec{v} dx'. \quad (5)$$

Ao invés de obter apenas a média na vizinhança x , pondera-se a RFO por uma função $h(x)$ que é normalmente a representação de um núcleo Gaussiano com a sua média nula e o seu desvio padrão s_x , conforme apresentado na Equação 6:

$$h(x) = \frac{1}{2\pi s_x^2} e^{-\frac{x^T x}{2s_x^2}}, \quad (6)$$

assumindo que:

$$f(x, t) = \int_{W(x)} h(x - x') f(x', t) dx' \quad (7)$$

e

$$S = \langle (\nabla_{x,t} I) (\nabla_{x,t} I)^T \rangle, \quad (8)$$

que é a definição do tensor de estrutura, a Equação 5 pode ser reescrita como:

$$\text{Argmin}_{\vec{v}} = (v_{x1}, v_{x2}, 1) \vec{v}^T S \vec{v}. \quad (9)$$

A minimização da função é um problema de médias quadradas ponderadas. Sendo assim, a solução pode ser encontrada ao considerar o sistema fornecido pela Equação 10.

$$\begin{pmatrix} \langle I_{x1}^2 \rangle & I_{x1} I_{x1} \\ I_{x1} I_{x2} & I_{x2}^2 \end{pmatrix} = \begin{pmatrix} \langle I_{x1} I_t \rangle \\ \langle I_{x2} I_t \rangle \end{pmatrix}. \quad (10)$$

A Figura 3(c) mostra o fluxo óptico calculado entre os *frames* e a Figura 3(d) apresenta o detalhe do fluxo óptico com o campo de vetores do movimento da perna.

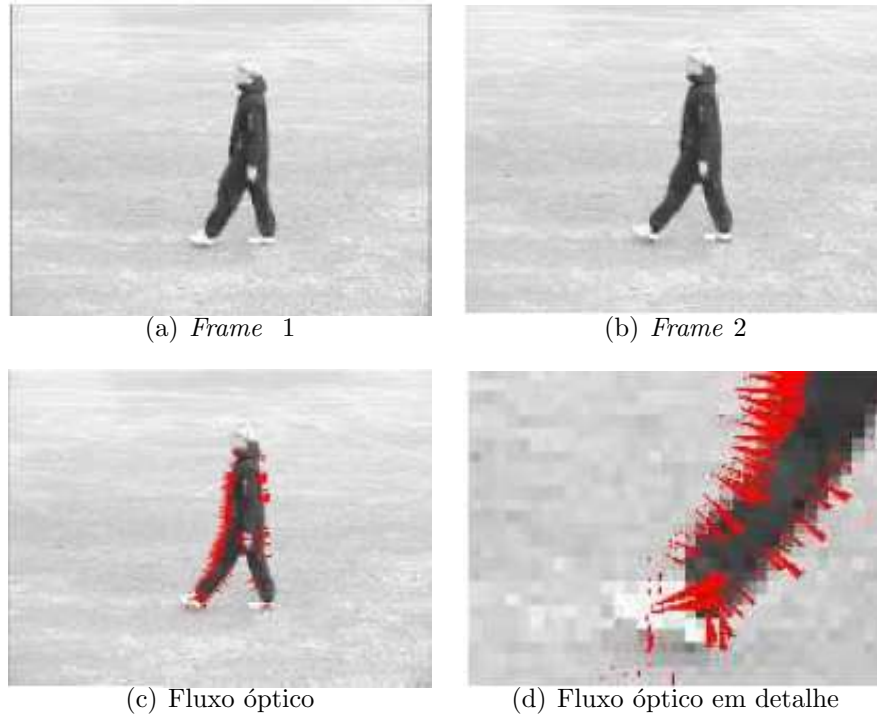


Figura 3: Fluxo óptico

1.1.2 Método de Horn e Schunck

O método de Horn e Schunck (HORN; SCHUNCK, 1981; BARRON; FLEET; BEAUCHEMIN, 1994) apresenta uma abordagem diferencial global para a extração do movimento. Com o objetivo de evitar variações no brilho devido ao efeito de sombras, assume-se que a superfície é plana e também que a iluminação incidente sobre a superfície é uniforme. Este método acrescenta ao cálculo do fluxo óptico uma restrição de coerência espacial do campo de deslocamento. Assim, temos que um campo de deslocamento pode ser calculado para resolver a Equação 3 de tal forma que as derivadas espaciais $\nabla \vec{v}$ sejam as mais fracas possíveis. Deste modo, minimiza-se a função definida na Equação 11:

$$\int \int_{\Omega} (\nabla I \cdot \vec{v} + I_t)^2 + \lambda^2 (\| \nabla v_{x_1} \|^2 + \| \nabla v_{x_2} \|^2) d_{x_1} d_{x_2}, \quad (11)$$

onde λ é um coeficiente que permite ponderar a influência da restrição de coerência.

O algoritmo proposto pelos autores é iterativo e converge para a solução. Para cada iteração i , o fluxo óptico é calculado conforme a Equação 12:

$$\begin{cases} v_{x_1}^{i+1} = v_{x_1}^{-i} - \frac{I_{x_1}(I_{x_1}v_{x_1}^{-i} + I_{x_2}v_{x_2}^{-i} + I_t)}{\alpha + I_{x_1}^2 + I_{x_2}^2} \\ v_{x_2}^{i+1} = v_{x_2}^{-i} - \frac{I_{x_2}(I_{x_1}v_{x_1}^{-i} + I_{x_2}v_{x_2}^{-i} + I_t)}{\alpha + I_{x_1}^2 + I_{x_2}^2} \end{cases}, \quad (12)$$

onde $\bar{v}_{x_1}^{-i}$ e $\bar{v}_{x_2}^{-i}$ são respectivamente a média dos vizinhos de v_{x_1} e v_{x_2} na iteração i , I_{x_1} e I_{x_2} são as derivadas espaciais em x_1 e x_2 de I e α é uma constante real. Esta nova restrição permite resolver o problema de abertura, mas não permite obter campos de deslocamento que apresentam descontinuidades.

1.1.3 Método de Augereau

Outro método, um pouco menos utilizado, é o método de Augereau (AUGEREAU; TREMBLAIS; FERNANDEZ-MALOIGNE, 2005). Este método calcula o fluxo óptico através de um método diferencial de estimação do movimento aparente e é fundado na hipótese da conservação da luminância dos *pixels* de uma imagem e na utilização de equações de derivadas parciais de filtragem direcional. Estas utilizam um operador diferencial chamado *tensor de estrutura*, permitindo determinar localmente, a partir de dados espaço-temporais, a direção do movimento aparente.

Sendo $\nabla I = (I_{x_1}, I_{x_2}, I_t)^T$ o gradiente de intensidades luminosas de um *pixel*, o tensor de estrutura S deste *pixel* é então uma matriz simétrica positiva $S\nabla I\nabla I^T$ é dada pela Equação 13:

$$S\nabla I\nabla I^T = \begin{pmatrix} I_{x_1}^2 & I_{x_1}I_{x_2} & I_{x_1}I_t \\ I_{x_1}I_{x_2} & I_{x_2}^2 & I_{x_2}I_t \\ I_{x_1}I_t & I_{x_2}I_t & I_t^2 \end{pmatrix}, \quad (13)$$

onde os elementos espectrais do tensor S são os autovalores β_i definidos na Equação 14:

$$\beta_1^{(S)} = I_{x_1}^2 + I_{x_2}^2 + I_t^2, \quad \beta_2^{(S)} = \beta_3^{(S)} = 0. \quad (14)$$

Levando em consideração os autovetores de S , o autovetor associado à $\beta_1^{(S)}$ é o vetor gradiente $v_1^{(S)} = \nabla I$. Além disso, o subespaço gerado pelos dois outros autovetores $v_2^{(S)}$ e $v_3^{(S)}$, associados aos autovalores nulos, é ortogonal à ∇I . Assim, qualquer vetor pertencente ao núcleo de S é uma possível solução para a hipótese de iluminação constante. De fato, $v_2^{(S)}$ e $v_3^{(S)}$ podem ser escolhidos de tal modo que $(v_1^{(S)}, v_2^{(S)}, v_3^{(S)})$ formem uma base ortogonal, conforme definido na Equação 15:

$$v_1^{(s)} = \begin{pmatrix} I_{x_1} \\ I_{x_2} \\ I_t \end{pmatrix}, \quad v_2^{(s)} = \begin{pmatrix} I_{x_2} \\ -I_{x_1} \\ 0 \end{pmatrix}, \quad v_3^{(s)} = \begin{pmatrix} I_{x_1}I_t \\ I_{x_2}I_t \\ -[I_{x_1}^2 + I_{x_2}^2] \end{pmatrix}. \quad (15)$$

Os métodos de Lucas e Kanade, Horn e Schunck e Augereau são considerados métodos diferenciais de extração de movimento e têm por princípio estudar a variação temporal das intensidades luminosas dos *pixels* na sequência de imagens. Para isso, assumem que a quantidade total de intensidade luminosa não varia entre *frames* adjacentes. Dessa forma, mesmo sendo considerado uma boa aproximação do movimento aparente, o cálculo do fluxo óptico ainda é sensível ao ruído e às mudanças bruscas de iluminação, e requer significativo poder computacional.

1.1.4 Outros Métodos

Para aplicações de vigilância em tempo real, calcular o fluxo óptico de cada *pixel* é computacionalmente dispendioso. O custo para calcular apenas o gradiente espaço-temporal é significativamente menor, mas requer informações de contorno. Em cenas com multidão, os indivíduos sobrepõem-se e suas posições mudam constantemente. Isso consequentemente dificulta a obtenção de informações sobre os contornos. Levando em consideração esses aspectos, a extração dos cantos dos objetos para representá-los em movimento minimiza o custo de computação (ZHAN et al., 2008). Desde a proposição do método de Lucas e Kanade, diversas melhorias foram apresentadas.

Em (BLACK; ANANDAN, 1993), é estabelecida uma nova estrutura baseada em estimativa para lidar com descontinuidades espaciais. A formulação global do fluxo óptico apresentada por Horn e Schunck depende da suposição de consistência de brilho e suavidade espacial, mas a sua formulação quadrática não é robusta para *outliers*. Black e Anandan superaram essa restrição particular modificando a função de erro quadrático para uma formulação mais robusta. O fluxo óptico de Black e Anandan não satisfaz as exigências para aplicações em tempo real, mas apresenta resultados mais precisos em relação ao método de Horn e Schunck.

Em (PAPENBERG et al., 2006), o método de Horn e Schunck (HORN; SCHUNCK, 1981) é melhorado para apreender grandes deslocamentos. Seu método lida melhor com as mudanças da condição de iluminação porque leva em consideração a variação de velocidade entre os *pixels* e a variação de luminância em diferentes instâncias. O método de Brox é uma forma alternativa para lidar com o desafio de proporcionar imunidade ao ruído introduzido pelas mudanças de iluminação.

1.2 Modelo de Força Social

O comportamento humano é caótico e imprevisível em situações complexas. No entanto, para situações relativamente simples, modelos estocásticos de comportamento podem ser desenvolvidos quando restringidos apenas ao comportamento probabilístico de um grupo. Mudanças comportamentais são guiadas pelo que se convencionou chamar de campos sociais ou forças sociais. O modelo de força social (MFS) proposto em (HELBING; MOLNAR, 1995) simula o fenômeno de auto-organização das pessoas na multidão (YAN; LI; MU-CHAO, 2012). No NFS, a força de interação entre pedestres é tratada como uma força social.

Quando um indivíduo está em uma situação considerada normal e previsível, sua reação é geralmente automática e determinada por experiências anteriores sobre como agir apropriadamente. Portanto, é possível apreender alguns aspectos do comportamento humano em modelos matemáticos. No modelo matemático que descreve o movimento de indivíduos, as mudanças temporais dv/dt da velocidade desejada $v_d(t)$ são descritas por uma quantidade vetorial $F(t)$ que pode ser interpretada como uma força social. Essa força representa o efeito do ambiente sobre o comportamento do indivíduo. No entanto, a força social não é exercida pelo ambiente no corpo do indivíduo. É, na verdade, uma quantidade estimada que descreve a motivação para agir. No caso do comportamento de pessoas, essa motivação evoca a produção física de uma aceleração ou desaceleração como reação ao estímulo percebido no ambiente (HELBING; MOLNAR, 1995).

No MFS, o comportamento do indivíduo é definido por três forças sociais: a força para a direção desejada, a força de interação com os pedestres da vizinhança e a força repulsiva exercida por obstáculos. A força pessoal F_{pes} para mover-se com a velocidade desejada v_{des} representa a tendência do pedestre de agir de acordo com sua vontade. A força de interação F_{int} com as pessoas ao redor e com o ambiente, impede que a vontade do pedestre seja concretizada em sua plenitude, porque que ele evita sempre esbarrar nas outras pessoas e em obstáculos concretos como paredes, postes, etc. (MEHRAN; OYAMA; SHAH, 2009). Assim, a força F_{pes} é atenuada pela força F_{int} .

O MFS descreve a dinâmica de movimento dos pedestres, considerando as motivações pessoais e as restrições do ambiente. Nesse modelo, cada pedestre i de massa m_i muda sua velocidade v_i^{actual} , de acordo com a Equação 16:

$$F_{res} = m_i \frac{dv_i}{dt} = F_{pes} + F_{int}, \quad (16)$$

onde F_{res} é a força resultante e consiste da soma de duas componentes: força F_{pes} associada ao desejo pessoal do indivíduo e a força de interação F_{int} , que está associada a interação com indivíduos próximos e com obstáculos.

Pessoas na multidão, enquanto se movem, geralmente definem um ponto de destino próximo. Por isso, é razoável considerar que cada indivíduo i tem uma direção e velocidade desejada v_i^{des} . Entretanto, a multidão limita o movimento individual e a velocidade corrente v_i^{atual} é diferente da velocidade desejada v_i^{des} . Os indivíduos tendem a aproximar sua velocidade desejada da velocidade atual com base na força pessoal F_{pes} .

$$F_{pes} = \frac{1}{\tau}(v_i^{des} - v_i^{atual}), \quad (17)$$

onde τ é um parâmetro de relaxação.

A força de interação F_{int} é composta da força de repulsão e atração F_{viz} , que representa a tendência psicológica em manter uma distância social entre os pedestres vizinhos, e da força ambiente F_{amb} que representa a tendência natural em evitar colisões com obstáculos. Portanto, a força de interação F_{int} é definida conforme a Equação 18:

$$F_{int} = F_{viz} + F_{amb}. \quad (18)$$

Os pedestres mantêm pequenas distâncias com pessoas conhecidas e distâncias grandes de indivíduos ou ambientes que lhe causam desconforto. No MFS, essas forças são definidas com base em funções de campos potenciais. O MFS generalizado pode ser resumido então conforme a Equação 19:

$$F_{res} = m_i \frac{dv_i^{atual}}{dt} = m_i \left(\frac{v_i^{des} - v_i^{atual}}{\tau} \right) + F_{int}. \quad (19)$$

1.3 Diferença Absoluta entre *Frames*

A subtração de plano de fundo, ou subtração de *background*, é um método comumente utilizado para realizar a segmentação de imagens nas situações em que o fundo é relativamente estático. A subtração de fundo consiste basicamente em selecionar ou então compor um *frame* que servirá como fundo da imagem, o qual não pode conter objetos em movimento ou qualquer outro tipo de objeto que não pertença ao fundo. Em seguida, é calculada a diferença *pixel a pixel* entre o *frame* atual e o de referência. Caso esta

diferença seja menor que um limiar predefinido, o *pixel* pertence ao plano de fundo, caso contrário, pertence ao objeto detectado (FERREIRA, 2007).

A detecção de primeiro plano pela subtração de plano de fundo é uma técnica de computação visual em que o primeiro plano de uma imagem é extraído para posterior reconhecimento de objetos. Geralmente, as regiões de interesse de uma imagem são objetos como seres humanos, carros, textos etc. A detecção de primeiro plano é uma metodologia amplamente utilizada para detectar objetos em movimento nos vídeos (KHILAR; CHITRAKALA, 2014).

A extração de primeiro plano é amplamente aplicada em aplicações de vigilância, como rastreamento de alvo, reconhecimento de objetos e análise de comportamento (WANG; WANG, 2014). O modelo de mistura gaussiana (GMM) (*Gaussian Mixture Model*) é o método mais representativo para a modelagem de fundo para detectar alvos no primeiro plano a partir de cenas estáticas (STAUFFER; GRIMSON, 1999).

Em (WANG; WANG, 2014), a extração de primeiro plano foi utilizada para detectar anormalidades em vídeos de curta duração. Uma sequência de símbolos é construída com base na série temporal da intensidade dos *pixels*, e em seguida, a transição entre os símbolos correspondentes à variância dos *pixels* ao longo do tempo é formulada utilizando o modelo de Markov, visando obter os limiares da detecção de anomalias.

Em (HASSAN et al., 2014), propõe-se um algoritmo adaptativo de modelagem de fundo para o sistema de vigilância de multidões. Nesse algoritmo, é utilizado o Método Mediano Aproximado (AMM) (*Approximate Median Method*) juntamente com um detector de borda de congruência de fase para desenvolver o modelo de fundo. Com base na extração de primeiro plano, o método proposto apresentou resultados que possibilitam aplicações de detecção em tempo real.

Em (SEESE et al., 2016), utilizou-se a extração de primeiro plano de imagens para realizar o reconhecimento de espécies de peixes embaixo da água. O objetivo era realizar a classificação com precisão e em tempo real utilizando unidades de processamento gráfico (GPU) (*Graphics Processing Unit*).

Em (BILAL; KHAN, 2015), é utilizada a obtenção de camadas de movimento pela diferença absoluta entre *frames* (DAF) para destacar as regiões com movimento. O algoritmo desenvolvido estima o movimento nas cenas utilizando a curva Gaussiana das magnitudes. Realiza-se um processamento paralelo em arranjos de portas programáveis

por campo (*Field Programmable Gate Array*) para implementar o sistema proposto. Os testes foram efetuados com imagens de câmera de segurança em tempo real.

Nesta dissertação, utilizou-se a diferença absoluta entre *frames* (DAF) (BILAL; KHAN, 2015) para realizar a extração de movimento dos vídeos. Trata-se de uma modificação do método de extração de plano de fundo, visando adequar as camadas de movimento geradas aos aspectos de implementação do algoritmo desenvolvido no Capítulo 5. A motivação para o uso dessa técnica provém da sua simplicidade e de implementações anteriores bem sucedidas na detecção de anormalidades em vídeos como em (WANG; WANG, 2014; HASSAN et al., 2014; SEESE et al., 2016). A DAF é calculada conforme a Equação 20:

$$C_{DAF}(x, y, k) = \sqrt{[F(x, y, k + 1) - F(x, y, k)]^2}, \quad (20)$$

onde $x = \{1, 2, \dots, \text{largura da imagem}\}$ e $y = \{1, 2, \dots, \text{altura da imagem}\}$ são as posições dos *pixels* no *frame* F e $C_{DAF}(x, y, k)$ é a camada contendo o movimento entre os *frames* k e $k + 1$.

1.4 Considerações Finais

Neste capítulo foram abordados alguns métodos de extração de movimento entre *frames*. O sistema proposto nesta dissertação extrai o movimento das cenas de 3 formas: utilizando o fluxo óptico obtido pelo método de Lucas e Kanade, utilizando o fluxo de forças obtido pelo MFS a partir do volume espaço-temporal do fluxo óptico e utilizando a técnica mais simples de diferença absoluta entre *frames*. O objetivo é realizar uma comparação da robustez das 3 técnicas.

Capítulo 2

TRABALHOS RELACIONADOS

ESTE capítulo apresenta alguns trabalhos relacionados ao problema da detecção e localização de anormalidades em cenas de multidão que apresentam alta densidade de pessoas. As soluções são organizadas nas Seções 2.1 e 2.2 segundo duas possíveis estratégias de detecção e localização: utilizando apenas fluxo óptico (FO) e utilizando o modelo de força social (MFS) computado a partir do FO. Na seção 2.3 são apresentadas algumas considerações finais.

2.1 Análise de Cenas por FO

O FO estima o movimento entre um par de *frames* de um vídeo (HORN; SCHUNCK, 1981). O princípio da técnica consiste em coincidir a probabilidade de movimento aparente entre os *frames* em relação às alterações da luminosidade dos *pixels*. A abordagem de FO foi usado em análise de cenas com multidão (WU; WONG, 2012), detecção anormalidades em vídeos (THIDA et al., 2012) e reconhecimento de expressões faciais (ANDERSON; MCOWAN, 2006).

Detecções de violência em vídeos de multidão podem ser realizadas pela observação de mudanças na covariância do FO, que tende a ser grande em tais casos, conforme apresentado em (HUANG; CHEN, 2014). Esse método considera as características estatísticas de campo do FO e extrai essas características através de um descritor SCOF (*Statistical Characteristic of the Optical Flow*), que passa então a representar a sequência de quadros do vídeo. O SCOF é caracterizado como normal ou anormal utilizando uma máquina de vetor suporte (SVM) (*Support Vector Machine*).

Para estimar a quantidade de pessoas em multidões densas que apresentam comportamento anormal (XIONG et al., 2012) desenvolveu um modelo que faz uso do FO. Nele,

uma distribuição probabilística de objetos no primeiro plano da imagem é obtida a partir de histogramas, que definem a entropia da aglomeração de pessoas. Um índice de distribuição da multidão é definido pelos resultados da contagem realizados com a entropia e passa então a representar a distribuição espacial das pessoas. A energia cinética é encontrada a partir do FO e do índice de distribuição. Por fim um limiar é utilizado para detectar o comportamento anormal nas cenas.

Em outra aplicação envolvendo análise de densidade de multidões e contagem de pessoas em vídeo, (WU; ZHENG; WANG, 2013) estima a densidade da multidão com base na análise de características regionais e pela regressão de vetor suporte (SVR) (*Support Vector Regression*). Nessa metodologia são extraídas características de cada segmento: taxa de *pixels* e blocos de histogramas do primeiro plano, dimensão de Minkowski das bordas da imagem e os níveis de cinza da matriz de co-ocorrência da imagem. A SVR é usada para treinar e estimar a densidade da multidão das regiões cujas características foram extraídas. Posteriormente, o valor de densidade estimado e o FO de Lucas e Kanade são usados para contar a quantidade de pedestres que ultrapassam uma linha virtual definida nos quadros do vídeo. Para isso, é feita uma divisão da estimativa de densidade da multidão pelo número de *pixels*, obtendo-se assim a densidade por *pixel* do primeiro plano. Então as velocidades dos pedestres baseadas no FO são usadas para computar o número de *pixels* do primeiro plano que ultrapassam a linha. A número estimado de pessoas atravessando a linha é resultado do produto da densidade por *pixel* do primeiro plano pela número de *pixels* do primeiro plano que atravessa a linha virtual.

Em (KIM et al., 2010) foi apresentado um outro método, mais simplificado, que fornece informações sobre a densidade de multidão e estima seu movimento utilizando o FO de Lucas e Kanade. O método é capaz de estimar o número de objetos se movendo a partir da quantidade de *pixels* nas bordas. A densidade de pessoas utiliza as duas informações estimadas e é classificada de acordo com uma quantidade de níveis de densidade definidos *a priori*.

Para detectar eventos em uma estrutura probabilística e interpretar automaticamente o comportamento da multidão em vídeos (RAO et al., 2015) utilizou *Manifolds* do FO para realizar a classificação. Foi proposto um algoritmo que detecta eventos em que pessoas caminham ou correm, fazendo uso do vetor de comprimento do FO. Além disso, foi proposto também um novo algoritmo para detectar concentrações e espalhamento de

peessoas, que faz uso das conexões de Riemannian no FO. Os maiores vetores encontrados fornecem as informações necessárias para distinguir eventos distintos, como pedestres caminhando ou correndo. Usando a conexão de Riemannian, os vetores do FO são também utilizados para localizar grupos específicos na multidão.

Em (LIU; LI; JIA, 2014) um novo método é introduzido para detectar aglomerado de pessoas movendo-se em destaque na multidão. Esse método é baseado na intensidade de todo o movimento da multidão que pode ser obtido pelo acumulo dos vetores do FO de um *frame* do vídeo. Desse modo, pode-se detectar atividade anormal no vídeo configurando um limiar pra detectar se a intensidade do movimento total mudou repentinamente. Entretanto, a computação do FO é sensível às mudanças nas condições de luminosidade da cena, o que pode resultar em falsas detecções. Por isso, para se adaptar as essas mudanças, um processo de treinamento é realizado com um conjunto de limiares estáticos, conhecidos a principio. Assim, esse método pode detectar comportamento anormal de forma robusta sem muito processamento computacional.

Um algoritmo baseado no FO foi utilizado em (EIBL; BRANDLE, 2008), e faz uso de métodos de agrupamento para encontrar campos de movimento dominante em trechos de vídeos de multidão. Esta formulação do FO foi capaz de preservar discontinuidades no campo, e se mostrou ser mais robusto contra mudanças de iluminação repentinas e oclusões. Nesta aplicação, utilizou-se *cluster* espectral com auto ajuste e isomapa de redução dimensional não-linear, combinado com *k-means*.

Em outra aplicação de agrupamento, (CHEN; HUANG, 2011) utilizou o FO para separar a multidão em grupos de forma não-supervisionada usando um método de aglomeração baseado na matriz adjacente (AMC) (*Adjacency-Matrix Based Clustering*). Com a divisão da multidão em grupos, os atributos como orientação, posição e tamanho são caracterizados por um modelo de campo de vetores. Isso torna possível predizer o comportamento da multidão baseado em modelos e assim detectar anormalidades presentes na cena.

2.2 Análise de Cenas por MFS

O modelo de força social para dinâmica de pedestres é parte da concepção de que o movimento dos pedestres pode ser descrito como estando sujeito à “forças sociais”. Essas “forças” não são exercidas diretamente pelos pedestres, são na verdade uma medida das

motivações internas dos pedestres de realizar certos movimentos. O modelo apresentado em (HELBING; MOLNAR, 1995) detalha os componentes destas forças sociais: um termo descreve a aceleração em direção à velocidade de movimento desejado, outro termo descreve a tendência do pedestre em manter uma certa distância dos outros pedestres e obstáculos e um terceiro termo modela o efeito de atratividade entre os indivíduos. A equação do movimento resultante é não-linear e foi inspirado na equação de Langevin que descreve o movimento Browniano em um potencial. O MFS é capaz de descrever a auto-organização de diversos efeitos coletivos observados em comportamento de pedestres de forma realística (XU et al., 2015).

2.2.1 Métodos Determinísticos

A abordagem do problema de análise de comportamento nas multidões utilizando MFS apresenta resultados similares ou superiores aos métodos que utilizam apenas o FO (ZHAN et al., 2008). Em (MEHRAN; OYAMA; SHAH, 2009) utiliza-se o MFS para detectar tumultos em vídeos com multidão. Nesse método, uma grade de partículas é colocada sobre a imagem e associada à média espaço-temporal do FO. As partículas são tratadas como indivíduos e as forças de interação estimadas pelo MFS. As forças de interações são então calculadas para obter o fluxo de forças (FF) de cada par de *frames* do vídeo. Volumes espaço-temporais são aleatoriamente selecionados para modelar o comportamento normal da multidão e um modelo de grupo de palavras classifica os *frames* como normais ou anormais. As regiões que apresentam anormalidades nos *frames* são localizadas através das forças de interação do FF.

Em (ZHAO et al., 2011), um novo método é proposto para realizar a análise e localização de instabilidades na multidão utilizando um campo de velocidades baseadas no MFS. Ao considerar o impacto do campo de velocidades nas forças de interação entre os indivíduos, melhorias no MFS foram introduzidas pela utilização da probabilidade de colisão baseada na distribuição de velocidades. Em comparação com o MFS proposto em (MEHRAN; OYAMA; SHAH, 2009), que define a força de interação como uma variável dependente da posição física dos indivíduos, o modelo melhorado em (ZHAO et al., 2011) é capaz de prover previsões melhores de interação usando a probabilidade de colisões na dinâmica da multidão. A partir da análise de instabilidades no espaço-temporal do vídeo, cenas com potenciais anormalidades podem ser extraídas, assim como as regiões de

interesse, em que as anormalidades no vídeos estão ocorrendo em determinadas cenas da sequência.

Uma modificação do MFS, considerando a velocidade relativa dos pedestres, é proposto em (GAO et al., 2013). Embora o MSF consiga descrever de modo realista o comportamento dinâmico dos pedestres, ainda existem aspectos do modelo que estão em desacordo com a realidade, como os casos em que pedestres se aproximam de uma parede com diferentes velocidades e a força repulsiva que é exercida sobre eles, que depende apenas das suas distâncias em relação à parede. Como resultado, nesses casos, um pedestre com alta velocidade pode experimentar uma colisão mais forte com a parede do que os outros. Entretanto, na vida real, o indivíduo com maior velocidade é capaz de prever essa colisão e reduzir sua velocidade rapidamente. Assim, (GAO et al., 2013) propôs um MFS modificado em que o pedestre está apto a prever possíveis colisões utilizando não somente informações sobre sua posição, mas também sobre sua velocidade. O pedestre prevê sua posição no momento seguinte com base na sua posição e velocidade no momento atual, e assim, a força de repulsão do pedestre é determinada pela posição prevista ao invés da posição corrente. Com essas modificações no modelo, a predição de colisões tornou-se mais realista.

O MFS pode ser usado não só para detectar anormalidades em cenas de multidão, mas também para simular o comportamento de multidão em situações específicas, como em (XU et al., 2015), em que o MFS foi utilizado para desenvolver uma nova técnica denominada *miSFM*, que simula ações de evacuação de multidões. A técnica *miSFM* promove o conhecimento global das atividades da multidão e incorpora um simulador com agentes que influenciam no comportamento coletivo do grupo. O modelo é construído com base no mapa do ambiente e sobre este mapa, o MFS é empregado para modelar o comportamento da multidão. A informação mútua (*mi*) entre os agentes serve de base para refinar os parâmetros utilizados no MFS. Mais especificamente, a informação mútua de interação dos agentes é adotada para determinar o nível de ordem na multidão durante a evacuação. A chave da inovação desta técnica encontra-se em evidenciar como os ajustes dinâmicos dos parâmetros do MFS revelam movimentos na multidão muito mais realistas para simulação de evacuação.

2.2.2 Métodos Estocásticos

O uso de inteligência de enxame produz consideráveis melhorias na técnica que utiliza FO e MFS, otimizando a detecção e localização de anormalidades nos vídeos com multidão. Ao realizar a rápida convergência do enxame de agentes para as áreas de interesse, ocorre uma forte diminuição no tempo exigido para análise de movimento entre *frames*. A versatilidade e capacidade de adaptação tornam o uso da inteligência de enxame ainda mais atrativa para lidar com o alto dinamismo e elevada instabilidade que o comportamento humano apresenta em tumultos.

Em (RAGHAVENDRA et al., 2011) a detecção de anormalidades em cenas de multidão é realizada utilizando otimização de enxame de partículas (PSO) (*Particle Swarm Optimization*), que otimiza as forças de interação computadas pelo MFS. Neste método, as partículas do enxame são direcionadas para as áreas de intenso movimento nos *frames* do vídeo, com as maiores magnitudes das forças de interação. O deslocamento das partículas é guiado pela função de aptidão, buscando maximizar as forças de interação para modelar o comportamento característico das pessoas no vídeo. A classificação dos volumes espaço-temporais é realizada pelo uso de um limiar relacionado ao somatório das forças de interação associadas as partículas em cada *frame* do vídeo.

Uma nova abordagem para detectar eventos anormais em cenas com multidão é apresentada em (KAL TSA et al., 2014). A teoria de enxame é aplicada para criação de características de movimento dos pedestres, através do histograma de aceleração de enxame orientado (HOSA) (*Histograms of Oriented Swarm Accelerations*), que é capaz de capturar eficientemente a dinâmica das cenas de multidão. O HOSA, juntamente com o histograma de gradientes orientados (HGO) (*Histograms of Oriented Gradients*), que descreve a aparência do quadro de vídeo, fornecem um descritor final baseado em movimento e aparência, para caracterizar as cenas. A aparência e movimento são extraídas de volumes espaço-temporais dos *pixels* que estão variando entre *frames*, ou seja, as regiões de interesse, permitindo que a detecção de anormalidades ocorra somente em pequenas regiões do vídeo. Em uma abordagem muito semelhante, (KAL TSA et al., 2015) utilizou o conceito de histogramas de enxames orientado (HOS) (*Histograms of Oriented Swarms*) aos invés de HOSA, para capturar a dinâmica de ambientes com aglomerações e tornar o método mais genérico.

2.3 Considerações Finais

Neste capítulo, foram apresentados os trabalhos relacionados ao problema de detecção e localização de anormalidades em cenas com elevada densidade de pessoas. Os métodos foram classificados segundo a utilização de 2 técnicas: FO e MFS. Os trabalhos relacionados revelam que ambas as técnicas são bem sucedidas na análise de cenas para detecção de eventos anormais nos vídeos. Os métodos que utilizam inteligência de enxame combinadas com o MFS apresentam uma redução do custo computacional envolvido na análise das cenas. Os trabalhos relacionados à diferença absoluta entre *frames* (DAF) foram apresentados anteriormente na Seção 1.3.

Capítulo 3

OTIMIZAÇÃO BASEADA NO FORRAGEAMENTO DE BACTÉRIAS

OS algoritmos de otimização por inteligência computacional consistem em métodos estocásticos capazes de resolver problemas de otimização numérica complexos, não-lineares e não-diferenciáveis. Esses problemas carecem de soluções analíticas que são muitas vezes encontradas através da busca exaustiva, uma técnica computacionalmente dispendiosa. Diferentemente da busca exaustiva, os algoritmos de inteligência computacional realizam a busca pela solução ótima utilizando técnicas adaptativas inspiradas na natureza. A busca adaptativa reduz a quantidade de avaliações porque privilegia apenas as melhores soluções de cada iteração. De uma forma geral os algoritmos de inteligência coletiva têm apresentado resultados melhores sob diversos aspectos, quando comparados às técnicas tradicionais (CIVICIOGLU, 2013).

3.1 Inteligência de Enxame

O termo inteligência de enxame refere-se às técnicas que utilizam algoritmos desenvolvidos a partir da simulação do comportamento social de sistemas vivos tais como colônias de bactérias, cardumes de peixes, enxame de insetos, bandos de aves, entre outros (BONABEAU; DORIGO; THERAULAZ, 1999). De uma forma geral, essas técnicas imitam a capacidade natural dos grupos de trabalhar em conjunto para achar a solução de problemas complexos que indivíduos isolados não conseguiriam encontrar. Nesses sistemas, agentes simples, dotados de pouca inteligência e recursos sensoriais muitas vezes limitados, compartilham suas informações com outros indivíduos do grupo, e estes, por sua vez, reajustam seus

comportamentos para a convergência do coletivo na direção de uma solução pseudo-ótima ou até mesmo ótima.

As formigas, por exemplo, são capazes de encontrar o caminho mais curto para a fonte de alimento com base em algumas regras simples de comportamento e pela comunicação baseada no acúmulo de feromônio na trilha (ENGELBRECHT, 2007). O comportamento das formigas na busca de alimentos inspirou o desenvolvimento do algoritmo de otimização por colônia de formigas, ou ACO (*Ant Colony Optimization*) (DORIGO; STÜTZLE, 2010), utilizado para otimizar problemas combinatórios como o do caixeiro viajante (BONABEAU; DORIGO; THERAULAZ, 1999).

Outro comportamento coletivo bem sucedido é aquele praticado pelas abelhas. As abelhas forrageiras (que buscam alimento), quando encontram uma boa área para extrair néctar, retornam para a colmeia e executam um conjunto de movimentos rítmicos na forma de dança. O propósito da dança é compartilhar informações sobre direção, distância e quantidade de néctar da fonte descoberta. O número de abelhas recrutadas para explorar a nova área é proporcional à dança executada. O comportamento coletivo das abelhas inspirou o algoritmo de colônias de abelhas artificiais (ABC) (*Artificial Bee Colony*) (KARABOGA et al., 2014).

Já a coreografia dos bandos de aves, cuja formação se assemelha a uma nuvem de insetos, inspirou o desenvolvimento do algoritmo de otimização por enxame de partículas (PSO) (SHI; EBERHART, 1998). Inicialmente, o algoritmo de otimização por enxame de pássaros foi desenvolvido para simular graficamente o comportamento social das aves, emulando os padrões que governam o sincronismo do voo entre elas. A partir daí, o modelo evoluiu para o PSO, que é mais simples e se adequa bem a problemas de otimização contínua.

Como outro notável exemplo, tem-se o algoritmo de otimização por forrageamento de bactérias (BFO) inspirado no comportamento social da *Escherichia coli*. As bactérias procuram nutrientes de forma a maximizar a energia obtida por unidade de tempo. O processo em que uma bactéria se move, dando pequenos passos a procura de nutrientes, é chamada de quimiotaxia. O algoritmo BFO imita o movimento quimiotático das bactérias reais para realizar buscas otimizadas (PASSINO, 2005) (PASSINO, 2012).

Nesta dissertação foi desenvolvida uma versão modificada e simplificada do algoritmo BFO. As modificações foram implementadas visando reduzir o processamento

computacional e também agregar ao algoritmo alguns aspectos comportamentais associados à formação de colônias de bactérias reais. O algoritmo BFO será descrito em detalhes ao longo do restante deste capítulo e o algoritmo de colônia de bactérias artificiais (CBA) será visto em detalhes no Capítulo 5.

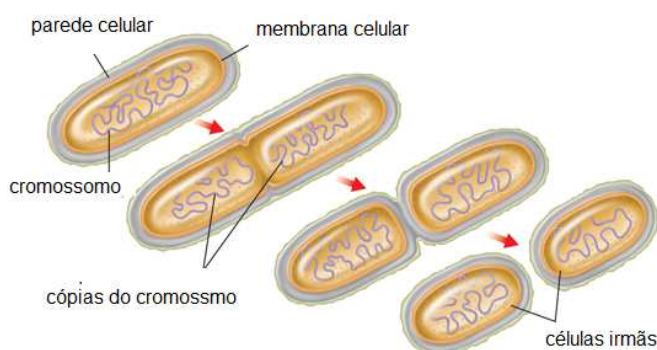
3.2 Bactérias Reais

Uma bactéria é um organismo unicelular procariótico. Seu DNA consiste unicamente de uma molécula circular. Sua estrutura é basicamente conformada por um corpo central de dimensões microscópicas que pode adotar formas diversas. Sua dimensão pode variar desde $0,01 \mu\text{m}^3$, como no caso da *Pelagibacter ubique*, até um tamanho 1010 vezes maior como no caso da *Epulopiscium fishelsoni*. Muitas bactérias são dotadas de flagelos rotativos na superfície, que funcionam como propulsores, e podem chegar a velocidade de até $30 \mu\text{m}$ por segundo (EISENBACH, 1990).

Além da estrutura apropriada para se movimentar, as bactérias contam com potentes receptores localizados na superfície celular, chamados de quimiorreceptores. Estes são capazes de detectar mudanças nas concentrações químicas do ambiente que os rodeiam. Dessa forma, quando detectam perturbações externas, mudam a sua locomoção de um movimento não dirigido para um movimento dirigido para uma direção específica. Para a sua sobrevivência, sintetizam permanentemente os sinais dos estímulos externos que lhe são vitais como o catabolismo do carbono, o fornecimento de nitrogênio, as localização das fontes de energia e o estado fisiológico da colônia (EISENBACH, 1990).

O tipo de reprodução mais comum nas bactérias é assexuada por bipartição ou cissiparidade. Neste processo, o DNA é replicado antes da divisão. Em seguida, a parede celular e a membrana plasmática crescem para dentro, dividindo a célula em duas. As bactérias multiplicam-se rapidamente por esse processo, quando dispõem de condições favoráveis (YOUNG, 2006).

Uma colônia microbiana é definida como um conjunto visível de microrganismos que crescem em uma superfície ou dentro de um meio sólido, presumivelmente cultivados a partir de uma única célula (TORTORA et al., 2004). Todos os microrganismos que dela descendem, assumindo que não houve nenhuma contaminação, são geneticamente idênticos, com exceção de quaisquer mutações que ocorram em baixas frequências. A obtenção de tais organismos geneticamente idênticos pode ser útil. Espalha-se organismos em uma



(a) Processo de divisão celular

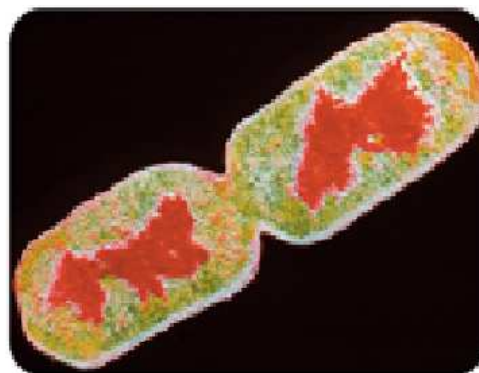
(b) Divisão da bactéria *Escherichia coli*

Figura 4: Reprodução bacteriana

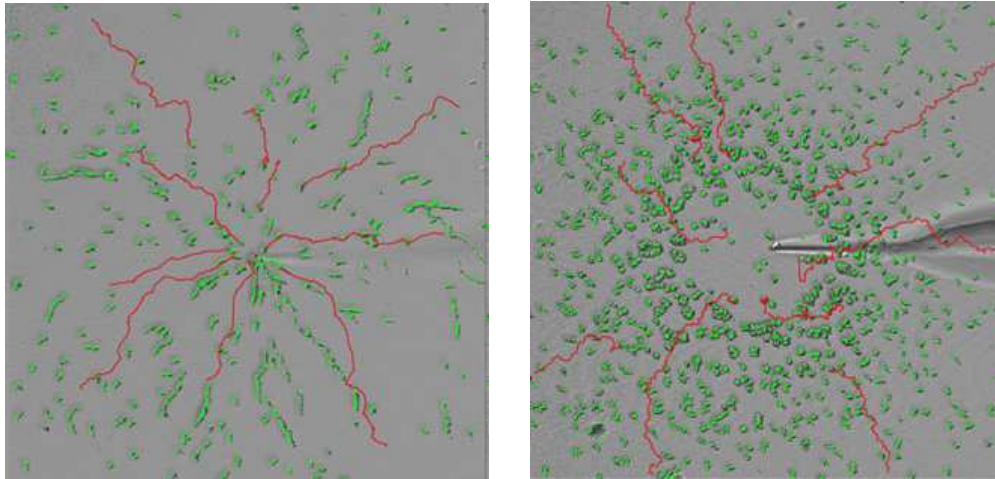
placa de cultura, iniciando um estoque novo a partir de uma única colônia resultante (YOUNG, 2006).

3.3 Quimiotaxia

Quimiotaxia é o movimento quimicamente dirigido que desenvolvem os seres vivos. Esse movimento dirigido e as substâncias químicas envolvidas nele são usados por alguns organismos unicelulares, insetos, mamíferos e até mesmo pelo homem, com diversos fins, como por exemplo, encontrar nutrientes, evitar predadores, estabelecer comunicação entre agentes na formação grupos, para atração sexual, ou para demarcar território (MURRAY, 2002). Na literatura científica, no entanto, o termo quimiotaxia é frequentemente usado para se referir ao movimento celular em resposta a gradientes de concentração de químicos presentes no ambiente. A Figura 5 ilustra a quimiotaxia.

A quimiotaxia faz parte da estratégia de sobrevivência das bactérias. Para crescerem, as bactérias precisam se alimentar; para se reproduzir, elas precisam se dividir; se as condições estiverem favoráveis onde se encontram, optam por permanecer; se as coisas estiverem melhores em outro lugar, preferem ir até lá; se forem ameaçadas, precisam fugir; e se o mundo ao redor mudar elas precisam mudar também (YOUNG, 2006).

Embora no universo bacteriano existam diversos mecanismos de locomoção, para este trabalho, os mecanismos relevantes são os sistemas quimiotáticos constituídos por flagelos. Os flagelos das bactérias têm forma helicoidal e podem girar a grandes velocidades (de 270 a 600 revoluções por segundo). Podem ainda se deter momentaneamente e mudar o sentido do giro de forma controlada (EISENBACH, 1990). Como consequência, quando



(a) Bactérias distribuídas em um meio de cultura

(b) Bactérias ao redor de uma concentração elevada de nutrientes

Figura 5: Ilustração da quimiotaxia

todos os flagelos giram no mesmo sentido, atuam como propulsores, movimentando a bactéria suavemente para frente em um movimento praticamente retilíneo. No caso da *Escherichia coli* e a da *Salmonella typhimurium*, os flagelos giram no sentido horário. Quando os flagelos giram na direção contrária à da natação, desestabilizam-se devido a sua forma helicoidal, impulsionando a célula em diversas direções. Assim o corpo da bactéria faz um giro aleatório. Para as bactérias que possuem flagelos, existem quatro estratégias quimiotáticas diferentes:

- Corrida e giros: a corrida consiste no deslocamento via movimentação dos flagelos e o giro é a reorientação estimulada por mudanças no gradiente de concentração dos nutrientes. Essa é uma estratégia extensamente pesquisada, entre outras coisas por ser a estratégia aplicada pela bactéria *Escherichia coli*.
- Corrida e arco: as bactérias reorientam sua direção girando um arco curto. A principal diferença entre essa estratégia e a anterior reside no fato de que a movimentação é produzida pela reorientação e não pela corrida. Até o momento tem-se observado esta estratégia unicamente na bactéria *Thiovulum majus*.
- Corrida e parada: a parada substitui o giro e o arco das estratégias anteriores. A reorientação acontece sob a ação de movimentos Brownianos. Essa estratégia é aplicada pela bactéria *Rhodobacter sphaeroides*.

- Corrida e reversa: geralmente as reversas são mais curtas do que as corridas e as bactérias movem-se em uma única direção. Costuma ser aplicada por bactérias marinhas.

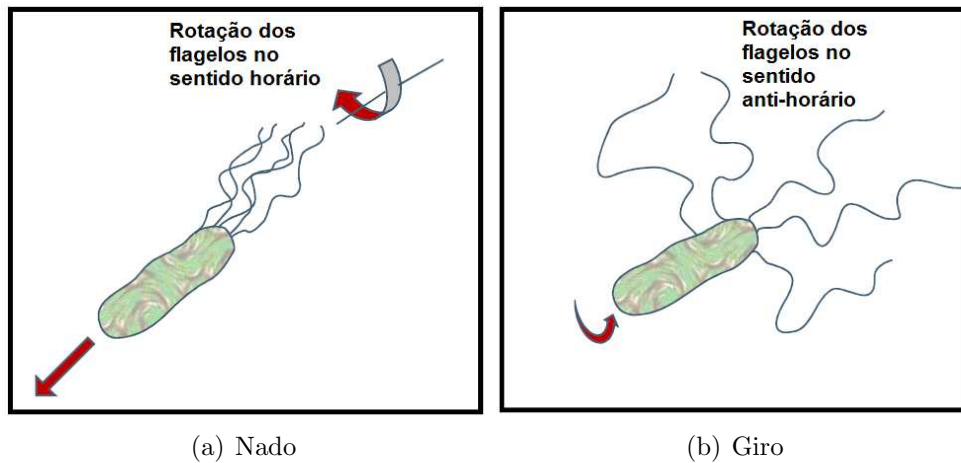


Figura 6: Movimento de nado e giro da bactéria *Escherichia coli*

Das quatro estratégias, a de corrida e giros é a mais conhecida e também a mais fácil de traduzir em regras. Como as bactérias normalmente estão inseridas em um meio fluido, o mais adequado seria utilizar o termo “nadar” ao invés de “correr”. Na figura 6 ilustra-se os movimentos da bactéria *Escherichia coli*.

As substâncias químicas presentes no ambiente servem como fonte de energia para as bactérias e são quimioatraentes. As substâncias nocivas que repelem a bactéria são quimiorrepelentes. Dependendo do tipo de substância e do gradiente, as respostas quimiotáticas de uma bactéria que aplica a estratégia natação-giro são:

1. Em ambientes neutros (nem atraente nem repelente), a bactéria alterna os movimentos de natação e giro, descrevendo um movimento aleatório.
2. Em ambientes quimioatraentes, mas sem gradiente, a longitude e velocidade média dos deslocamentos por nado aumentam, e a duração dos giros diminui.
3. Caso encontre o gradiente quimioatraente ou quimiorrepelente, a bactéria alterna os movimentos de natação e giro. Se o estímulo for positivo (gradiente positivo no caso do quimioatraente, e negativo no quimiorrepelente), a probabilidade de giros diminui; no caso de um estímulo negativo, essa probabilidade aumenta, e desta forma, a bactéria busca os estímulos positivos e foge dos estímulos negativos.

3.4 Algoritmo de Otimização por Forrageamento

O algoritmo de otimização por forrageamento de bactérias (BFO) proposto em (PASSINO, 2005) inspirou-se no comportamento de forrageamento da bactéria *Escherichia coli*. A biologia subjacente à estratégia de forrageamento da *Escherichia coli* é emulada de forma extraordinária e usada em processos de otimização. Recentemente o algoritmo BFO tem atraído a atenção de pesquisadores devido à sua elegância e leveza na otimização de funções multi-dimensionais.

Supondo que seja necessário encontrar o mínimo da função $J(\theta)$, $\theta \in \mathbb{R}^p$, e não haja uma descrição analítica do gradiente $\nabla J(\theta)$, o uso de técnicas inspiradas no forrageamento das bactérias apresenta-se como uma alternativa para a solução do problema. O algoritmo BFO imita os quatro principais mecanismos observados nas bactérias reais: quimiotaxia, formação de colônias, reprodução e eliminação-dispersão.

A bactéria virtual representa uma possível solução para o problema e, ao mesmo tempo, se comporta como um agente de busca, movendo-se na superfície da função a procura do ótimo global. A Figura 7 ilustra o modo como as bactérias artificiais realizam o forrageamento na função de custo.

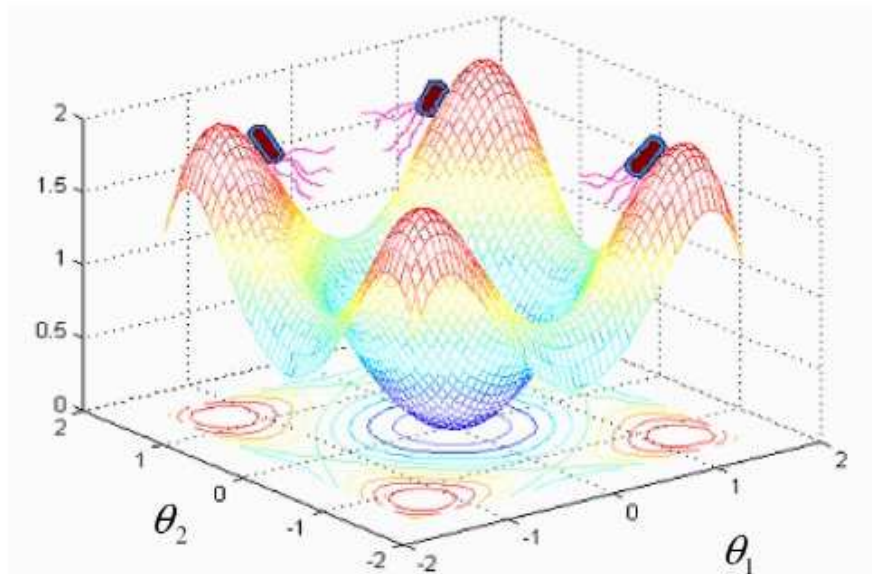


Figura 7: Quimiotaxia na função de custo

Quanto ao passo quimiotático, pode ocorrer primeiramente um giro seguido por outro giro ou um giro seguido por um deslocamento para frente. O índice j é adotado para

se referir ao passo quimiotático. Do mesmo modo, define-se k para passo de reprodução e l para a eliminação-dispersão. Ademais

p : define a dimensão do espaço de busca,

S : representa o número total de bactérias da população,

N_C : é o número de passos quimiotáticos,

N_s : representa a quantidade nadados,

N_{re} : indica o número de passos reprodutivos,

N_{ed} : corresponde ao número de eventos de eliminação-dispersão,

N_{ped} : refere-se a probabilidade da bactérias ser submetida a eliminação-dispersão,

$C(i)$: é o comprimento do movimento realizado após o giro.

Deste modo, $P(j, k, l) = \theta^i(j, k, l) | i = 1, 2, \dots, S$ representa a posição de cada bactéria da colônia S e $J(i, j, k, l)$ é o custo da posição da i -ésima bactéria $\theta^i(j, k, l) \in \mathbb{R}^p$. Note que $J(\theta)$ é tanto referida como custo, usando a terminologia da teoria de otimização, e como superfície de nutrientes, em referência ao conceito biológico. Abaixo são descritos os quatro comportamentos básicos do algoritmo BFO.

1. Quimiotaxia: esse processo simula o movimento de nado e giro das bactérias no fluido com nutrientes. Ela pode nadar por um período de tempo na mesma direção ou pode girar e depois recomeçar a nadar. Supondo que $\theta^i(j, k, l)$ representa a i -ésima bactéria no j -ésimo passo quimiotático, no k -ésimo passo reprodutivo e l -ésima eliminação-dispersão.

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (21)$$

onde Δ indica o vetor na direção aleatória, entre $[-1,1]$.

2. Formação de colônia: as bactérias se agrupam quando liberam substâncias que realizam a atração ou repulsão das vizinhas, apresentando um comportamento característico de enxame e formando padrões concêntricos com alta densidade de bactérias. Nos casos em que o BFO leva em consideração a comunicação entre as bactérias pode ser representado por:

$$J_{cc}(\theta, P(j, k, l)) = \sum_{i=1}^S J_{cc}(\theta, \theta^i(j, k, l)) \quad (22)$$

$$J_{cc}(\theta, P(j, k, l)) = \sum_{i=1}^S \left[-d_{atrac}^{-w_{atrac} \sum_{m=1}^p (\theta_m - \theta_m^i)^2} \right] + \sum_{i=1}^S \left[-h_{repul}^{-w_{repul} \sum_{m=1}^p (\theta_m - \theta_m^i)^2} \right] \quad (23)$$

onde $J_{cc}(\theta, P(j, k, l))$ é valor que deve ser adicionado à função objetivo a ser otimizada, para evitar variações temporais e $\theta = [\theta_1, \theta_2, \dots, \theta_p]^T$ é um ponto na área de busca p -dimensional. d_{atrac} , w_{atrac} , h_{repul} e w_{repul} são parâmetros que definem o grau de atração e repulsão entre as bactérias, respectivamente.

3. Reprodução: as bactérias menos saudáveis morrem enquanto as mais saudáveis (que têm magnitudes baixas na função objetivo) reproduzem assexuadamente, dividindo-se em duas bactérias, posicionadas nas proximidades do local, onde ocorreu a reprodução.
4. Eliminação-dispersão: grupos de bactérias em regiões localizadas em áreas da função distantes do ótimo local morrem e são aleatoriamente reposicionadas em outra região do espaço de busca. O balanço entre eliminação-dispersão e reprodução se mantém constante.

O Algoritmo 1 apresenta os passos seguidos na otimização pelo BFO. Em seguida, apresenta-se também uma descrição pormenorizada das etapas.

Para inicialização do algoritmo são configurados os parâmetros p , S , N_c , N_s , N_{re} , N_{ed} , N_{ped} , e $C(i), i = 1, 2, \dots, S$. As posições iniciais são aleatoriamente distribuídas pelo domínio da função. O algoritmo que modela a quimiotaxia, reprodução e eliminação-dispersão iniciam com $j = k = l = 0$. As atualizações para o θ^i resultam automaticamente em atualizações para $P(j, k, l)$. Seguem os passos do algoritmo BFO.

[Passo 1] Inicialização dos parâmetros p , S , N_c , N_s , N_{re} , N_{ed} , N_{ped} , $C(i)$ e θ^i .

[Passo 2] *loop* de eliminação-dispersão: $l = l + 1$.

[Passo 3] *loop* de reprodução: $k = k + 1$.

[Passo 4] *loop* de quimiotaxia: $j = j + 1$.

[a] Para $i = 1, 2, \dots, S$ realizar o passo quimiotático da bactéria i como segue.

[b] Computa a aptidão $J(i, j, k, l)$ na função.

[c] Salva em $J_{ultimo} = J(i, j, k, l)$ o último valor até que um valor melhor seja encontrado.

Algoritmo 1 Algoritmo BFO**Entrada:** função de aptidão J ;**início**Inicialia $p, S, N_c, N_s, N_{re}, N_{ed}, N_{ped}, C(i)$ e θ^i **repita****repita****repita****para** $i=1$ até S **faça**Computa a aptidão $J(i, j, k, l)$ na função;Salva em $J_{ultimo} = J(i, j, k, l)$;Gera um vetor ramdômico de giro $\Delta(i) \in \mathfrak{R}^p$;Realiza um deslocamento $\theta^i(j+1, k, l)$ $= \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$;Computa a aptidão $J(i, j+1, k, l) = J(i, j+1, k, l)$;Inicia o nado com $m=0$;**repita** $m=m+1$;**se** $J(i, j+1, k, l) < J_{ultimo}$ **então** $J_{ultimo} = J(i, j+1, k, l)$;**fim**Realiza um deslocamento $\theta^i(j+1, k, l)$ $= \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$;**até** $m < N_s$;**fim****até** $j < N_c$; $J_{saudavel} = \sum_{j=1}^{N_c+1} J(i, j, k, l)$;

Duplicação das bactérias que obtiveram menor acúmulo de nutrientes durante a busca;

até $k < N_{re}$;Eliminação das piores bactérias de acordo com a probabilidade P_{ed} , mantendo a população constante;**até** $l < N_{ed}$;**fim****Saída:** $[J_{ultimo}]$;

[d] Giro: gera um vetor ramdômico $\Delta(i) \in \mathfrak{R}^p$ em que cada elemento $\Delta_m(i), m = 1, 2, \dots, p$ é um número aleatório que pertence ao intervalo $[-1, 1]$.

[e] Comprimento do deslocamento:

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (24)$$

Esse resultado é um deslocamento de tamanho $C(i)$ na direção do giro da bactéria i .

[f] Computa $J(i, j + 1, k, l)$:

$$J(i, j + 1, k, l) = J(i, j, k, l) \quad (25)$$

[g] Início dos nados para a bactéria i :

i) Para $m = 0$ (contador do número de deslocamentos)

ii) Enquanto $m < N_s$ e ainda não tiver alcançado o ótimo da função.

Fazer $m = m + 1$

Se $J(i, j + 1, k, l) < J_{ultimo}$, fazer $J_{ultimo} = J(i, j + 1, k, l)$ e

$$\theta^i(j + 1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (26)$$

utilizando $\theta^i(j + 1, k, l)$ para computar o novo $J(i, j + 1, k, l)$ como feito em [f]

[h] Ir para a próxima bactéria ($i + 1$) se $i \neq S$ (i.e., ir para [b] para processar a próxima bactéria).

[Passo 5] Se $j < N_c$, ir para o passo 4 e continuar desde que todas as bactérias não tenham morrido.

[Passo 6] Reprodução:

[a] Para um dado k and l , para cada $i = 1, 2, \dots, S$

$$J_{saudavel}^i = \sum_{j=1}^{N_c+1} J(i, j, k, l), \quad (27)$$

onde $J_{saudavel}^i$ é a medida de quantos nutrientes adquiriu e quão bem sucedida foi em evitar substâncias nocivas.

[b] As piores bactérias morrem e as restantes reproduzem assexuadamente. Nesse processo é realizada a duplicação da bactéria no mesmo local.

[Passo 7] Se $k < N_{re}$, ir para o passo 3.

[Passo 8] Eliminação-dispersão: para $i = 1, 2, \dots, S$ com probabilidade P_{ed} , eliminar e dispersar as bactéria piores bactérias, de modo a manter constante a população. Para isso, se uma bactéria é eliminada, outra é criada em uma posição aleatória do espaço de busca. Se $l < N_{ed}$, ir para o passo 2 ou finalizar o algoritmo.

A Figura 8 ilustra o comportamento das bactérias nas curvas de nível da função de duas dimensões “Sphere”, mostrada na Equação 28. É possível observar nessa figura que já na segunda geração houve uma rápida convergência das bactérias para o mínimo da função.

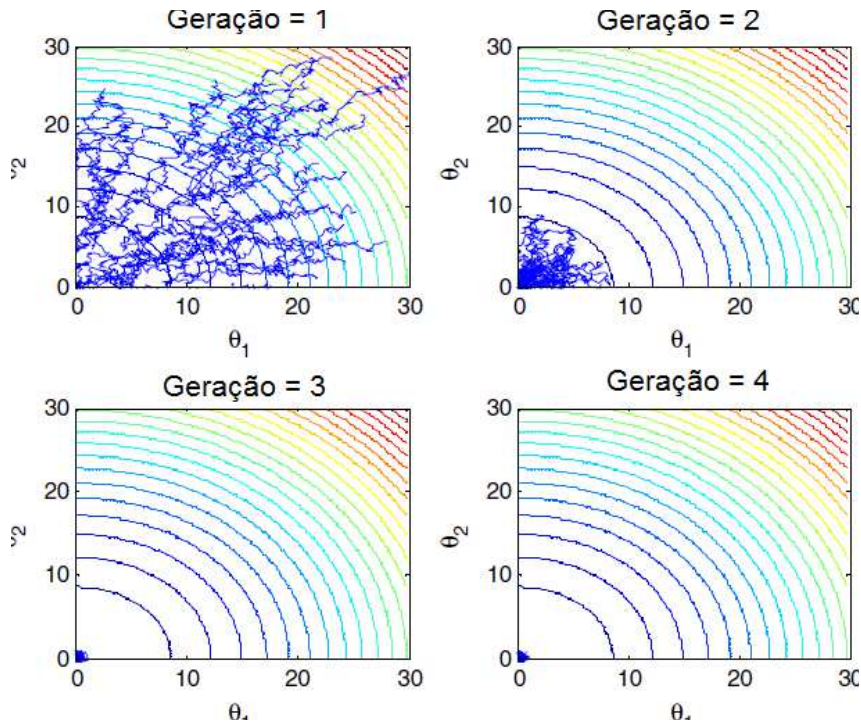


Figura 8: Comportamento de convergência das bactérias virtuais nas curvas de nível da função *Sphere* (PASSINO, 2005).

$$f(x_1, x_2) = x_1^2 + x_2^2. \quad (28)$$

3.5 Aplicações

Nas áreas de bioquímica, biologia, genética entre outras, a modelagem da quimiotaxia de bactérias e sua aplicação formam problemas amplamente estudados. Contudo, somente na última década vem-se publicando trabalhos científicos com aplicações em engenharia. Nesta seção, apresenta-se um apanhado de trabalhos publicados relevantes e aplicados a diversos campos da engenharia.

Em (KIRCHNER; SCHADSCHNEIDER, 2002), inspirado no processo de quimiotaxia, realiza-se uma modelagem e simulação da formação de rotas de fuga, baseando-se no estudo dos processos de evacuação simples em casos de catástrofes, usando os conceitos de autômatos celulares estocásticos para modelar a dinâmica de pedestres, de modo que se possa determinar e classificar os comportamentos manifestados durante a simulação. Conseguiu-se desenvolver um modelo de simulação de alta velocidade devido à simplicidade técnica, aplicável na otimização de procedimentos de evacuação.

Em (MARTONE et al., 2007) aplicam-se diferentes algoritmos heurísticos de otimização na solução de problemas de engenharia elétrica, relacionados com sistemas de potência e de energia. Entre os algoritmos aplicados está o BCA (*Bacterial Chemotaxis Algorithm*) (MULLER et al., 2002). Os algoritmos heurísticos são testados e seu desempenho comparado, revalando a robustez da técnica.

Como se evidencia, a quimiotaxia é uma estratégia otimizada de movimentos. Em (BREMERMAN, 1974) é realizada uma das primeiras referências ao mecanismo da quimiotaxia das bactérias, cujo algoritmo de otimização considera a concentração não homogênea de quimioatraentes. No algoritmo, as bactérias realizam os seguintes passos:

1. Gira e nada em linha reta em uma direção aleatória.
2. Continua até que o gradiente comece a decrescer, caso essa direção seja a de um gradiente positivo do quimioatraente. Depois de medir o decrescimento, a bactéria gira novamente e, como resultado, nada em uma nova direção aleatória.
3. Para e depois gira por um tempo e nada em outra direção, caso a nova direção seja a de um gradiente decrescente do quimioatraente.
4. Assume-se que as direções aleatórias sucessivas são independentes e uniformemente distribuídas.

Em (MONTAGUE; WARD, 1994) aplica-se um algoritmo inspirado no trabalho de (BREMERMAN, 1974) na otimização de biorreatores. Utilizou-se este algoritmo para encontrar o perfil de alimentação, a concentração inicial e a adição adequada de enzimas que maximiza a função de desempenho de reatores.

Em (LIU; PASSINO, 2002) apresentou-se o conceito de métodos de otimização distribuída não baseados em gradiente. Analisou-se a quimiotaxia das bactérias *Escherichia coli* e *Myxococcus xanthus*. Para a *Escherichia coli*, apresenta-se novamente o algoritmo BFO, mas para a bactéria *Myxococcus xanthus* desenvolveu-se um novo algoritmo do tipo autômato celular estocástico, no qual as bactérias podem se locomover em um espaço tridimensional, e nele o comportamento coletivo das bactérias recebe mais importância. Posteriormente em (GAZI; PASSINO, 2002), desenvolveram-se validações de convergência e análise de estabilidade do algoritmo BFO.

O algoritmo original e algumas de suas versões também têm sido aplicados com sucesso na solução de diversos problemas de engenharia. A seguir mencionam-se alguns

destes trabalhos. Em (MISHRA, 2005) é apresentada uma aplicação para a estimação dos harmônicos em sistemas de potência. Em (KIM; CHO, 2006), o algoritmo BFO é utilizado na afinação de um controlador PID (*Proportional Integral Derivative*) de um AVR. Em (LI et al., 2007), utiliza-se a quimiotaxia para encontrar a solução ótima para um OPF (*Optimal Power Flow*). Em (LIN; LIU; FANG, 2007), são estimados os parâmetros das funções e os pesos de um conjunto de regras de um sistema de lógica nebulosa utilizando BFO. Em (FANG; LIU; LIU, 2007), é implementada a estimação dos parâmetros de um sistema de modelagem não-linear do fenômeno de fratura, utilizando estratégias de forrageamento bacteriano. Em (PANIGRAHI; PANDI, 2008), o algoritmo BFO é utilizado para solucionar problemas do tipo *economic load dispatch* e finalmente em (GUNEY; BASBUG, 2008), é utilizado no aperfeiçoamento de uma aplicação para arranjos lineares de antenas no controle da amplitude dos elementos.

3.6 Considerações Finais

Este capítulo apresentou algumas técnicas de otimização inspiradas em inteligência de enxame e abordou em detalhes o algoritmo BFO. Nesta dissertação as modificações do algoritmo BFO são apresentadas no Capítulo 5. Tais modificações foram motivadas pela necessidade de tornar o algoritmo BFO mais simples e leve, possibilitando assim aplicações de baixo custo computacional. Alterações também foram realizadas para adaptar os conceitos fundamentais do BFO ao problema abordado nesta dissertação, que não consiste apenas em maximizar a função de custo, mas sim posicionar as bactérias artificiais em todas as regiões da função cujos valores estão acima de um limiar dinâmico.

Capítulo 4

REDES NEURAIS DE KOHONEN

GRANDE parte das arquiteturas de redes neurais artificiais utilizam um conjunto de padrões de entrada e saída para realizar o treinamento. Porém, em algumas aplicações, somente o conjunto de entrada é disponibilizado, inexistindo nesses casos, as respectivas saídas desejadas. As amostras de entrada, no entanto, em alguns casos carregam consigo informações relevantes sobre o comportamento esperado do sistema. Para essas aplicações, o uso de redes neurais auto-organizáveis costuma ser uma alternativa acertada. Os pesos sinápticos dessas redes se ajustam por meio de métodos de treinamento competitivo, os quais têm a capacidade de detectar as similaridades, regularidades e correlações entre os dados do conjunto de entrada, agrupando-os em classes (*clusters*). Cada classe está relacionada com situações e condições que governam o processo. Assim, as classes revelam as relações intrínsecas existentes entre os elementos do conjunto de treinamento (SILVA; SPATTI; FLAUZINO, 2010). Esta seção apresenta os fundamentos teóricos do método de treinamento competitivo e da arquitetura da rede neural de Kohonen (RNK), também conhecida como SOM (*Self-Organizing Map*). A RNK foi utilizada nesta dissertação para realizar a classificação dos registros comportamentais das colônias de bactérias artificiais geradas durante a otimização das camadas de movimento dos vídeos.

4.1 Aprendizado Competitivo

Para descrever os passos envolvidos no processo de aprendizado competitivo, utilizados no treinamento da RNK, considera-se uma estrutura constituída de apenas uma camada de neurônios, conforme ilustra a Figura 9.

O princípio básico que rege o aprendizado competitivo é a concorrência entre os neurônios. Cada neurônio tem como propósito sair vitorioso da competição. Vence o

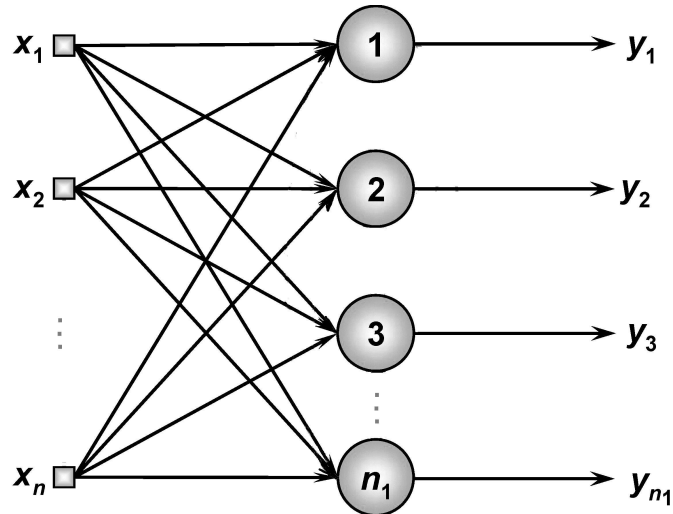


Figura 9: Estrutura neural básica da rede competitiva

neurônio cujo vetor de pesos está mais próximo do vetor de entrada. A recompensa para o neurônio vencedor é o ajuste dos seus pesos visando uma aproximação ainda maior da amostra na entrada. O vetor de pesos referente ao neurônio j , onde $j = 1, \dots, n_1$, é então representados como

$$\begin{aligned} w^{(1)} &= [W_{1,1} W_{1,2} \dots W_{1,n}]^T \\ w^{(2)} &= [W_{2,1} W_{2,2} \dots W_{2,n}]^T \\ w^{(3)} &= [W_{3,1} W_{3,2} \dots W_{3,n}]^T \\ &(\dots) \\ w^{(n_1)} &= [W_{n_1,1} W_{n_1,2} \dots W_{n_1,n}]^T. \end{aligned}$$

Uma das medidas mais utilizadas para identificar o neurônio vencedor consiste em determinar a proximidade existente entre o vetor de pesos de cada neurônio e o vetor de entrada $x^{(k)}$ contendo os elementos da k -ésima amostra. Uma métrica de proximidade comumente usada é a distância Euclidiana entre os dois vetores, ou seja:

$$dist_j^{(k)} = \sqrt{\sum_{i=1}^n (x_i^{(k)} - w_i^{(j)})^2}, \quad (29)$$

onde $j = 1, \dots, n_1$ e $dist_j^{(k)}$ é a distância Euclidiana entre o vetor de entrada e o vetor de pesos do j -ésimo neurônio $w^{(j)}$.

Assim, o neurônio $w^{(j)}$ que tiver o vetor de pesos mais próximo do vetor de entrada $x^{(k)}$, será considerado o vencedor da competição. Como recompensa pela vitória, o vetor de pesos do neurônio vencedor é ajustado no sentido de se aproximar ainda mais da amostra $x^{(k)}$. Para isso, utiliza-se a seguinte método de adaptação:

$$w_{\text{atual}}^{(v)} = w_{\text{anterior}}^{(v)} + \eta \cdot (x^{(k)} - w_{\text{anterior}}^{(v)}) \quad (30)$$

onde v é o número do neurônio vencedor, $w^{(v)}$ denota o vetor de pesos desse neurônio vencedor e o parâmetro η define a taxa de aprendizagem.

Visando uma maior eficiência no processo de aprendizagem, os vetores do conjunto de treinamento e dos pesos são normalizados. A normalização unitária é comumente utilizada nesses casos e efetua-se dividindo cada vetor pelo seu respectivo módulo (SILVA; SPATTI; FLAUZINO, 2010).

A Figura 10 ilustra a regra de ajuste de pesos para o neurônio vencedor, em que três neurônios, representados por $w^{(1)}$, $w^{(2)}$ e $w^{(3)}$, disputam entre si durante apresentação da amostra $x^{(k)}$.

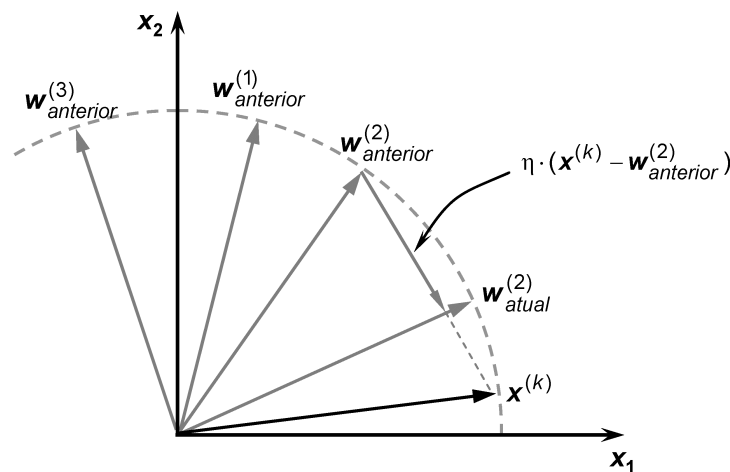


Figura 10: Ajuste de pesos do neurônio vencedor

Na Figura 10, com os vetores já normalizados, é possível observar que o neurônio $w^{(2)}$ vence a competição, porque está mais próximo do padrão $x^{(k)}$. Então $w^{(2)}$ será ajustado conforme a Equação 30, para que fique ainda mais próximo de $x^{(k)}$. Em resumo, o ajuste consiste em rotacionar o neurônio vencedor em direção ao vetor da amostra.

É importante ressaltar que a quantidade inicial de neurônios a ser utilizado, objetivando representar as classes com características em comum, é de antemão desconhecido, pois o aprendizado é não-supervisionado. Assim, é fundamental a obtenção de informações adicionais a respeito do problema a ser mapeado, seja por meio de especialistas ou por meio de métodos estatísticos, capazes de proporcionar uma estimativa inicial sobre a quantidade de possíveis classes associadas ao respectivo problema.

Como exemplo de um problema com quatro classes, a Figura 11 ilustra a distribuição dos vetores de pesos da rede após o treinamento. As amostras do problema são constituídas de vetores contendo apenas dois elementos: x_1 e x_2 .

Na Figura 11, os vetores ficam localizados dentro do círculo unitário porque são bidimensionais e estão unitariamente normalizados. A estrutura da rede possui quatro neurônios que correspondem aos quatro vetores de pesos $w^{(1)}$, $w^{(2)}$, $w^{(3)}$ e $w^{(4)}$. Os vetores de pesos se posicionaram próximos aos núcleos dos aglomerados de dados.

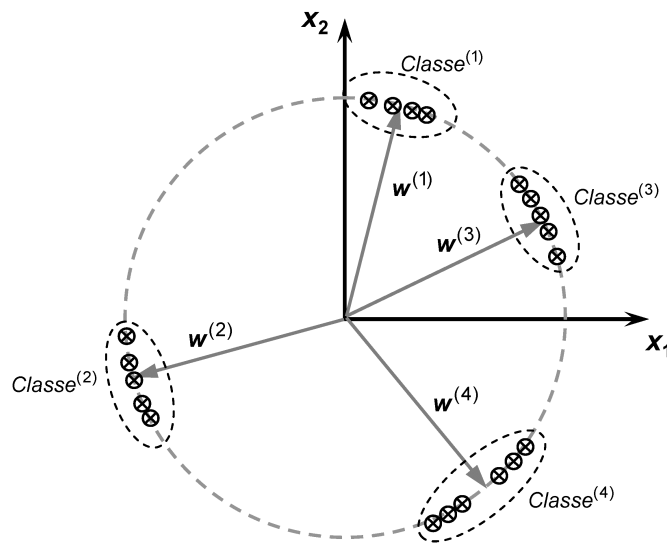


Figura 11: Representação dos vetores de peso

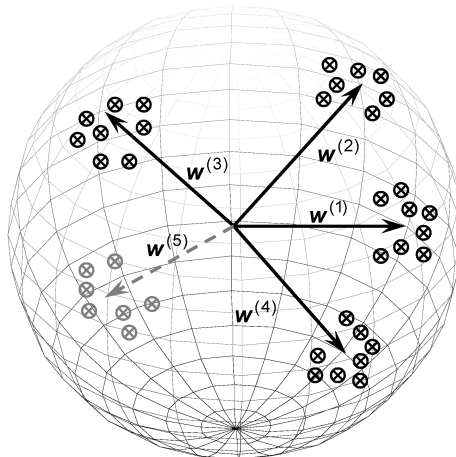


Figura 12: Distribuição tridimensional dos vetores de pesos em relação ao conjunto de amostras (SILVA; SPATTI; FLAUZINO, 2010)

A Figura 12 ilustra uma aplicação contendo três dados de entrada, semelhante a aplicação realizada nesta dissertação. Com a normalização, os vetores de amostras e

de pesos ficam confinados numa esfera de raio unitário. Mais uma vez é possível notar que os 5 neurônios, representados pelos seus correspondentes vetores de pesos $w^{(1)}$, $w^{(2)}$, $w^{(3)}$, $w^{(4)}$ e $w^{(5)}$, estão posicionados próximos ao centro dos cinco agrupamentos de dados referentes ao mesmo problema.

O Algoritmo 2 apresenta os passos seguidos durante a aplicação do método de treinamento competitivo. Visando primeiramente a garantia da posterior convergência do algoritmo, as diversas variáveis de entrada devem necessariamente ser linearmente independentes, de tal modo que as suas projeções ocorram dentro de uma hipersfera. Em um caso contrário, a superfície deixa de ser esférica porque os seus respectivos eixos constituintes não são mais dotados de ortogonalidade. Isso é alcançado com a normalização dos dados de entrada, e a configuração inicial dos pesos a partir de uma proximidade com um número pequeno número de amostras para ajuste.

Algoritmo 2 Algoritmo Competitivo – Fase de Treinamento

Entrada: conjunto de amostras \mathbf{x} ; número de épocas n_e ;

início

Inicia o vetor de pesos de cada neurônio considerando os valores das n_1 primeiras amostras de treinamento;

Normaliza os vetores de peso;

Normaliza o conjunto de amostras \mathbf{x} ;

Defini a taxa de aprendizagem η ;

época $\leftarrow 0$;

repita

para $k=1$ até o número total de amostras **faça**

$$dist_j^{(k)} = \sqrt{\sum_{i=1}^n (x_i^{(k)} - w_i^{(j)})^2};$$

 Declarar como vencedor $w^{(v)}$ o neurônio $w^{(j)}$ com a menor distância;

$$w_{atual}^{(v)} = w_{anterior}^{(v)} + \eta \cdot (x^{(k)} - w_{anterior}^{(v)});$$

fim

 Incrementa o contador de épocas;

até época $\leq n_e$;

fim

Após a convergência do algoritmo e com a rede devidamente treinada, para cada nova amostra de entrada, basta verificar o neurônio vencedor e a qual classe ele pertence.

O Algoritmo 3 apresenta os passos do realizados na fase de operação da rede neural treinada pelo método competitivo. Não há alterações dos vetores de pesos sinápticos durante a fase de operação da rede, pois tais ajustes são somente realizados durante a fase de treinamento.

Algoritmo 3 Algoritmo Competitivo – Fase de Operação**Entrada:** conjunto de amostras \mathbf{x} ;**início**Normaliza \mathbf{x} ;Assume os vetores de pesos $w^{(j)}$ obtidos após o treinamento;**para** $k=1$ até o número total de amostras **faça**

$$dist_j^{(k)} = \sqrt{\sum_{i=1}^n (x_i^{(k)} - w_i^{(j)})^2};$$

Declara como vencedor $w^{(v)}$ o neurônio $w^{(j)}$ com a menor distância;

Associa a amostra à classe representada pelo neurônio vencedor;

fim**fim**

4.2 Redes Neurais de Kohonen

A RNK é um modelo de rede neural que utiliza o método de treinamento não-supervisionado e competitivo para agrupar e modelar grandes quantidades de dados com baixo custo computacional. As RNK têm sido utilizadas com sucesso em diversas áreas e aplicações.

O algoritmo de treinamento foi originalmente proposto em (KOHONEN, 1982), e agrupa os padrões de entrada de alta dimensão, em uma ou duas dimensões. Isso é feito de tal modo que os neurônios localizados próximos uns dos outros, na camada de saída da rede, possuem padrões de entrada semelhantes. A rede de Kohonen é não-linear e tem a capacidade de preservar a estrutura topológica dos dados (VESANTO; ALHONIEMI, 2000).

As RNK encontram inspiração biológica no córtex cerebral onde a ativação de uma região específica corresponde à resposta frente a determinados estímulos sensoriais, como estímulo motor, visual ou auditivo. Embora a rede de Kohonen seja utilizada para diversas aplicações, em diferentes áreas do conhecimento, os trabalhos mais bem disseminados versam sobre problemas que envolvem classificação de padrões e agrupamento de dados (VESANTO; ALHONIEMI, 2000).

As redes de Kohonen, em essência, são estruturas neurais como aquela apresentada na Figura 9. Os mapas topológicos informam como estão organizados espacialmente os neurônios da rede frente ao comportamento de seus vizinhos e normalmente são formados por uma ou duas dimensões. A Figura 13 mostra um mapa topológico de apenas uma dimensão, em que os neurônios estão organizados na forma de uma única linha. Assim como apresentado na Figura 9, o vetor de entrada x que corresponde a amostra do conjunto de treinamento, será devidamente apresentado a entrada e a todos os neurônios da rede neural de Kohonen.

Os mapas podem também apresentar uma configuração de duas dimensões. Para realizar a ilustração de um mapa bidimensional, arranjado na forma de linhas e colunas, constituindo-se de 81 neurônios no total, é apresentado em detalhes na Figura 14.

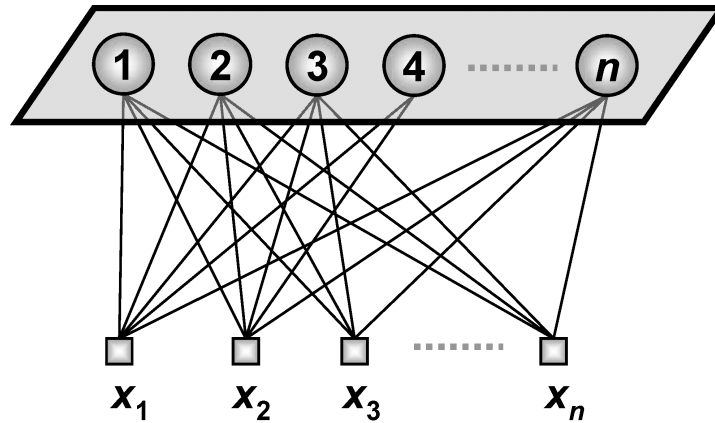


Figura 13: Mapa topológico unidimensional da RNK

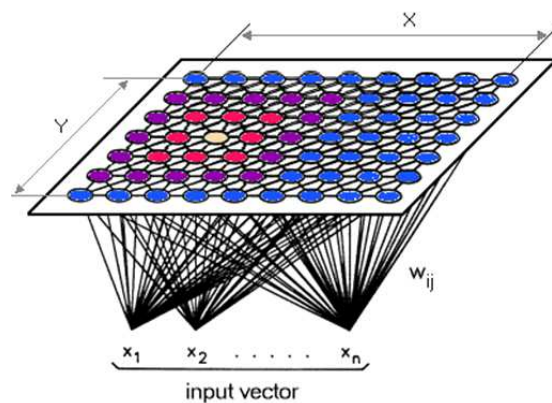


Figura 14: Mapa topológico bidimensional da RNK

O critério de vizinhança caracteriza a interação entre o neurônio e seus vizinhos. Um dos critérios de vizinhança mais utilizados consiste em especificar o raio R de abrangência, utilizado pelos neurônios para definir seus respectivos vizinhos. Deste modo, para um determinado neurônio j , seus vizinhos são todos aqueles que estão a uma distância máxima que seja menor ou igual a R .

Quando um neurônio j vence a competição após a apresentação de uma amostra na entrada, tanto o seu vetor de pesos como os de seus vizinhos são ajustados. Todavia, os neurônios que estão na vizinhança do neurônio vencedor são ajustados com taxas menores àquelas usadas para o ajuste do vetor de pesos do neurônio vencedor. Para regiões de vizinhança que estejam a uma distância maior do que R , a taxa de aprendizado costuma

ser ponderada por uma função do tipo Gaussiana, e quanto maior for a distância do neurônio vizinho em relação ao neurônio vencedor, menor será o ajuste dos pesos. Nessas condições a Equação 30 divide-se em duas regras:

$$\begin{cases} \text{Regra 1: } w_{\text{atual}}^{(v)} = w_{\text{anterior}}^{(v)} + \eta \cdot (x^{(k)} - w_{\text{anterior}}^{(v)}) \\ \text{Regra 2: } w_{\text{atual}}^{(\Omega)} = w_{\text{anterior}}^{(\Omega)} + \eta \cdot \alpha^{(\Omega)} \cdot (x^{(k)} - w_{\text{anterior}}^{(\Omega)}), \end{cases} \quad (31)$$

onde Ω está associado aos vizinhos do neurônio vencedor e $\alpha^{(\Omega)}$ é um operador de vizinhança dado por:

$$\alpha^{(\Omega)} = e^{\left(-\frac{\|w^{(v)} - w^{(\Omega)}\|^2}{2\sigma^2}\right)}. \quad (32)$$

A aplicação do operador Gaussiano permite uma taxa de decaimento que fica bem intensificada para os neurônios mais afastados do neurônio vencedor. A Figura 15 ilustra esse decaimento da taxa de aprendizado e, assim, podemos observar que os neurônios na vizinhança do neurônio vencedor, dentro da área de raio R , serão ajustados com uma taxa de aprendizado maior do que os neurônios vizinhos fora da área de raio R . Adicionalmente, conforme relatado em (RITTER et al., 1992), uma outra estratégia utilizada para aumentar a eficiência do processo de ajuste dos pesos, consiste em diminuir gradativamente a taxa de aprendizado no decorrer do treinamento.

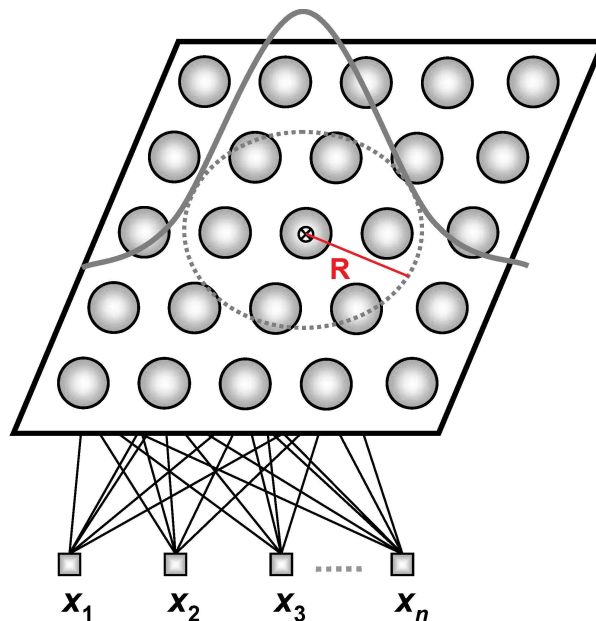


Figura 15: Ilustração do decaimento Gaussiano para o ajuste dos pesos dos neurônios vizinhos ao vencedor

O Algoritmo 4 apresenta os passos seguidos durante o treinamento da RNK. A única diferença entre o Algoritmo 4 e o Algoritmo 2 encontra-se na aplicação da regra de ajuste dos pesos. Para o Algoritmo 4 existe uma regra específica apenas para o ajuste dos pesos do neurônio vencedor e uma outra regra para o ajuste dos pesos dos neurônios vizinhos com base no operador Gaussiano.

Algoritmo 4 Algoritmo de treinamento da RNK

Entrada: Conjunto de treinamento \mathbf{x} ; taxa de aprendizado η ; número de épocas de treinamento n_e ;

início

Iniciar os vetores de pesos de cada neurônio $w^{(j)}$ considerando os valores das n_1 primeiras amostras de treinamento;

Normaliza os vetores dos pesos e das amostras do conjunto de treinamento;

época $\leftarrow 0$;

repita

para $k=1$ até o número total de amostras **faça**

$$dist_j^{(k)} = \sqrt{\sum_{i=1}^n (x_i^{(k)} - w_i^{(j)})^2};$$

Declara como vencedor $w^{(v)}$ o neurônio $w^{(j)}$ com a menor distância Euclidiana;

Ajusta os pesos do vencedor conforme a regra 1:

$$w_{atual}^{(v)} = w_{anterior}^{(v)} + \eta \cdot (x^{(k)} - w_{anterior}^{(v)});$$

Ajusta os pesos dos neurônios vizinhos conforme regra 2:

$$w_{atual}^{(\Omega)} = w_{anterior}^{(\Omega)} + \eta \cdot \alpha^{(\Omega)} \cdot (x^{(k)} - w_{anterior}^{(\Omega)});$$

fim

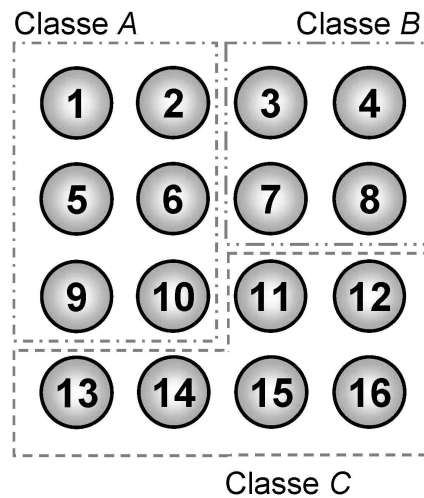
Incrementa o contador de épocas;

até até que época $\leq n_e$;

fim

Como RNK não possui de antemão as saídas desejadas, a identificação das regiões que podem ser agrupadas em classes distintas requer ferramentas estatísticas ou conhecimento especializado. Deste modo o mapa topológico pode ser particionado em regiões que definem as respectivas classes, sendo tal arranjo chamado de mapa de contexto. A Figura 16 exemplifica um mapa de contexto para uma topologia bidimensional após o treinamento.

Quando uma nova amostra é apresentada à RNK, basta identificar qual foi o neurônio vencedor da competição e em seguida recorrer ao mapa de contexto para resgatar a classe correspondente a este neurônio. O Algoritmo 5 exhibe os passos da fase de operação da rede de Kohonen após o treinamento. A principal diferença entre o Algoritmo 5 e o Algoritmo 3 está na utilização do mapa de contexto na classificação da amostra de entrada



(a) Mapa de contexto para 16 neurônios

Figura 16: Exemplo de um mapa de contexto para a topologias bidimensional

na fase de operação do Algoritmo 5.

Algoritmo 5 Algoritmo de operação da RNK

Entrada: conjunto de amostras \mathbf{x} a ser classificado;

início

Normaliza o conjunto \mathbf{x} ;

Assume os valores dos pesos $w^{(j)}$ ajustados durante a fase de treinamento;

para $k=1$ até o número total de amostras **faça**

$$dist_j^{(k)} = \sqrt{\sum_{i=1}^n (x_i^{(k)} - w_i^{(j)})^2};$$

Declara como vencedor o neurônio $w^{(j)}$ que contenha a menor distância Euclidiana;

Localiza o neurônio vencedor;

Associa a amostra à classe que foi identificada a partir do mapa de contexto;

fim

fim

A estrutura e configuração simples, assim como a dinâmica de treinamento diferenciada, fazem da RNK uma ferramenta sofisticada para aplicações em problemas que envolvem classificação de padrões e agrupamentos. O algoritmo de aprendizado competitivo tem forte semelhança com o algoritmo *k-means* (STEINHAUS, 1956), em que as amostras disponíveis devem ser divididas em k grupos (representados pelos seus centros geométricos), tendo-se como o critério de alocação a norma Euclidiana entre a amostra e os referidos centros.

4.3 Considerações Finais

Este capítulo apresentou os aspectos mais relevantes do método de treinamento competitivo utilizado no processo de aprendizagem da RNK. Nesta dissertação, a RNK foi escolhida devido a relativa simplicidade da sua arquitetura, pela inspiração biológica no córtex cerebral e pela natureza competitiva do método de treinamento. A RNK é utilizada aqui para classificar o conjunto de dados gerados durante a otimização das camadas de movimento pelas CBA.

Capítulo 5

ASPECTOS DE IMPLEMENTAÇÃO

NESTE capítulo são apresentados os aspectos de implementação do sistema de detecção de eventos em vídeos. A Seção 5.1 apresenta a arquitetura do sistema proposto e a Seção 5.2 descreve o modo como as técnicas de extração de movimento foram utilizadas neste trabalho. A Seção 5.3 apresenta o algoritmo de otimização CBA, a Seção 5.4 mostra como foi realizada a implementação da RNK e por último, na Seção 5.5, são apresentadas algumas considerações finais a respeito do capítulo.

5.1 Arquitetura do Sistema

O sistema proposto extrai o movimento entre *frames* utilizando 3 técnicas diferentes para comparar o tempo de processamento requerido por cada uma. A diferença absoluta entre *frames* (DAF) apresentada na Seção 1.3, o fluxo óptico (FO) apresentado na Seção 1.1 e o fluxo de forças (FF) apresentado na Seção 1.2, são as técnicas utilizadas para gerar as camadas de movimento. As camadas são então tratadas como funções de custo e otimizadas por um algoritmo inspirado no BFO e na habilidade natural das bactérias de se agruparem em colônias. O registro comportamental das bactérias artificiais durante a otimização compõe o conjunto de treinamento da RNK. Após o treino, a rede torna-se capaz de separar em classes os eventos distintos ocorridos no vídeo. A Figura 17 ilustra os estágios do sistema desenvolvido.

Na entrada do sistema, todos os *frames* do vídeo são convertidos de RGB para escala de cinza, reduzindo a imagem a apenas uma única matriz de cores, ao invés de 3. As técnicas de DAF, FO e FF extraem o movimento entre *frames* consecutivos e geram camadas que contém as magnitudes de variação de brilho dos *pixels*. A extração de

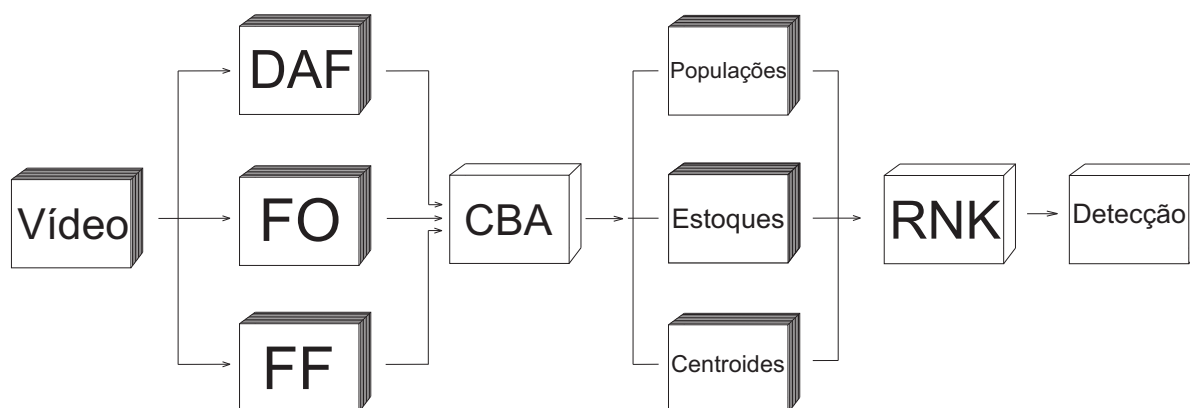


Figura 17: Diagrama de blocos do sistema proposto

movimento de uma sequência de *frames* gera os volumes espaço-temporais. As camadas do volume espaço-temporal são tratadas como funções de custo, em que os maiores valores da função são as regiões de interesse, porque estão associadas à presença de significativo movimento de objetos entre os *frames*. O algoritmo de otimização recobre as regiões de interesse com colônias de bactérias artificiais. Na primeira época de cada camada, a população inicial de bactérias é espalhada aleatoriamente sobre a camada de movimento. O ciclo de vida de cada bactéria mimetiza o ciclo de vida dos organismos sociais submetidos às leis de seleção natural Darwiniana, em que somente os mais adaptados conseguem sobreviver e se reproduzir. As bactérias menos aptas, impossibilitadas de obter nutrientes durante o forrageamento, sucumbem à eliminação. A camada com o movimento entre *frames* é, deste modo, o ambiente virtual que estimula a competição entre as bactérias.

Os vetores contendo os registros das colônias de bactérias que emergiram ao longo da otimização formam o conjunto de treinamento da RNK. O método de treinamento competitivo da rede agrupa em classes as colônias que apresentam padrões de comportamento semelhantes. Após o treinamento, classes numéricas são associadas aos grupos de padrões, e nesta parte do sistema proposto, a referência aos trabalhos anteriores e o reconhecimento de eventos pelo ponto de vista humano, realizam a associação entre a classe na saída da RNK e o evento ocorrido no vídeo. A partir deste momento, o sistema está habilitado para realizar a detecção de eventos automaticamente.

5.2 Extração de Movimento

Esta seção descreve como as técnicas de extração de movimento foram utilizadas nesta dissertação para obter o movimento entre *frames* e gerar os volumes espaço-temporais.

5.2.1 Diferença Absoluta entre *Frames*

A DAF, como descrito na Seção 1.3, gera camadas de movimento pela subtração direta de *frames* consecutivos, *pixel* por *pixel*. Usa-se o módulo da diferença para que a otimização lide apenas com valores positivos. A Equação 33 mostra como a camada de movimento é gerada a partir de um par de *frames*:

$$C_{DAF}(x_{DAF}, y_{DAF}, k) = \sqrt{[F(x_p, y_p, k + 1) - F(x_p, y_p, k)]^2}, \quad (33)$$

onde $k = \{1, 2, \dots, \text{total de frames} - 1\}$ e está associado ao número da camada do volume espaço-temporal gerado. C_{DAF} é a camada de movimento k obtida a partir dos *frames* consecutivos $F(x_p, y_p, k)$ e $[F(x_p, y_p, k + 1)]$. $x_{DAF} = \{1, 2, \dots, \text{número de colunas}\}$ e $y_{DAF} = \{1, 2, \dots, \text{número de linhas}\}$ são as posições das magnitudes correspondentes a largura e o comprimento da camada, respectivamente. Assim como $x_p = \{1, 2, \dots, \text{número de colunas}\}$ e $y_p = \{1, 2, \dots, \text{número de linhas}\}$ são as posições dos *pixels* no *frame*, também correspondentes à largura e altura.

A DAF tem a vantagem de ser extremamente simples e exigir muito pouco processamento computacional, mas é completamente susceptível a ruídos provenientes da baixa qualidade do vídeo e das mudanças repentinas nas condições de iluminação da cena. As magnitudes do ruído, no entanto, quando comparadas aos elevados valores das magnitudes das regiões com movimento, não apresentam atratividade nutricional para as bactérias artificiais. Sendo assim, durante a otimização realizada pelo algoritmo CBA, a quantidade de ruído não exerce uma influência proibitiva ao uso da DAF. Esse foi um dos fatores que motivou a utilização desta técnica no sistema proposto. Tal aspecto da implementação será avaliado com maiores detalhes no Capítulo 6.

5.2.2 Fluxo Óptico

O fluxo óptico é a distribuição de velocidades aparentes do movimento dos padrões de brilho em uma imagem. Consequentemente, o FO pode fornecer informações importantes sobre a disposição espacial dos objetos e a taxa de alteração desta disposição. A extração

de movimento realizada neste trabalho utiliza o FO obtido pelo método de Lukas e Kanade (LUCAS; KANADE et al., 1981) apresentado na Seção 1.1. O método de Lukas e Kanade assume que a velocidade aparente dos padrões de brilho varia suavemente entre dois *frames* consecutivos.

Nesta dissertação utilizou-se o *optical flow toolbox* do MATLAB[®], gratuitamente disponibilizado em (KARLSSON, 2016). A função *FlowLK* apresenta o FO melhorado pela aplicação local da regularização de Tikhonov (GROETSCH, 1984), que visa estabilizar o algoritmo.

A camada de movimento utilizada na otimização realizada pelas CBA consiste no módulo do campo de vetores das velocidades aparentes obtidas pelo método de Lukas e Kanade conforme a Equação 34:

$$C_{FO}(x_{FO}, y_{FO}, k) = \sqrt{u(x_u, y_u, k)^2 + v(x_v, y_v, k)^2}, \quad (34)$$

onde $k = \{1, 2, \dots, \text{total de frames} - 1\}$ e está associado ao número da camada do volume espaço-temporal do FO. $C_{FO}(x_{FO}, y_{FO}, k)$ é a camada de movimento k obtida a partir do módulo das velocidades aparentes $u(x_u, y_u, k)$ e $v(x_v, y_v, k)$, e x_u e y_u são as posições das magnitudes de u , assim como x_v e y_v são as posições das magnitudes de v .

5.2.3 Fluxo de Forças

Em (MEHRAN; OYAMA; SHAH, 2009) e (RAGHAVENDRA et al., 2011) utiliza-se as camadas do volume espaço-temporal do FO para obter o fluxo de forças (FF) pelo modelo de força social (MFS). O módulo dos vetores do FF representam a magnitude da força social calculada sobre as variações das magnitudes dos padrões de brilho provenientes do FO. Assim, as forças sociais são associadas aos *pixels* do *frame* ao invés de indivíduos do vídeo como um todo.

Para obter o FF a partir do FO, calcula-se primeiramente o FO $O(x_i, y_i)$ na posição (x_i, y_i) da imagem. Esse fluxo é então associado à velocidade atual v_i^{atual} conforme a Equação 35:

$$v_i^{atual} = O(x_i, y_i). \quad (35)$$

De fato, para evitar instabilidades numéricas, o valor da magnitude do FO $O(x_i, y_i)$ é um valor médio calculado para uma pequena vizinhança espacial. A velocidade desejada

v_i^{des} é associada a intensidade média $O_{med}(x_i, y_i)$ do FO na posição (x_i, y_i) dos *frames* anteriores ao *frame* corrente, conforme Equação 36:

$$v_i^{des} = O_{med}(x_i, y_i). \quad (36)$$

Finalmente, calcula-se a força de interação F_{int} usando a Equação 37:

$$F_{int}(x_i, y_i) = m_i \frac{dv_i^{atual}}{dt} - \frac{m_i}{\tau} (v_i^{des} - v_i^{atual}), \quad (37)$$

onde m_i é uma constante, τ é um parâmetro de relaxação e $\frac{dv_i^{atual}}{dt}$ é a diferença da magnitude do FO entre o *frame* corrente (t) e o anterior ($t-1$), com exceção do primeiro. Assim tem-se a Equação 38:

$$\frac{dv_i^{atual}}{dt} = O(x_i)|_t - O(x_i)|_{(t-1)}. \quad (38)$$

Conforme observado a partir da Equação 37, a força de interação representa a mudança no movimento do indivíduo do caminho desejado para o real. Quanto mais regular o movimento dos pedestres, menor as forças de interação, porque que o FF varia suavemente. Assim, em um cenário de aglomeração normal, espera-se que a força de interação se estabilize em torno de um valor característico para a cena. Durante tumultos, este valor se distancia rapidamente do limiar que caracteriza a normalidade.

O algoritmo para calcular o FF desenvolvido nesta dissertação utiliza os valores de u e v das velocidades aparentes do FO para obter os valores F_{int_u} e F_{int_v} das forças de interação. As camadas do FF são geradas pelo conforme a Equação 39:

$$C_{FF}(x_{FF}, y_{FF}, k) = \sqrt{F_{int_u}(x_u, y_u, k)^2 + F_{int_v}(x_v, y_v, k)^2}, \quad (39)$$

onde $k = \{1, 2, \dots, \text{total de frames} - 1\}$ e está associado ao número da camada do volume espaço-temporal do FF. $C_{FF}(x_{FF}, y_{FF}, k)$ é a camada de movimento k obtida a partir do módulo das forças de interação F_{int_u} e F_{int_v} . Por fim, x_u, y_u, x_v e y_v são as posições dos valores das forças de interação.

A Figura 18 mostra um exemplo de camadas obtidas pela DAF, FO e FF para o mesmo par de imagens consecutivas. É possível notar que a DAF revela apenas o contorno das formas que estão em movimento, enquanto o FO e o FF preenchem o objeto. Nota-se também que há um pouco menos de ruído no FF em relação ao FO. Quanto à camada

obtida pela DAF, embora haja pouco ruído presente neste exemplo em questão, a técnica é bastante vulnerável à instabilidade nas condições de luminosidade na cena.

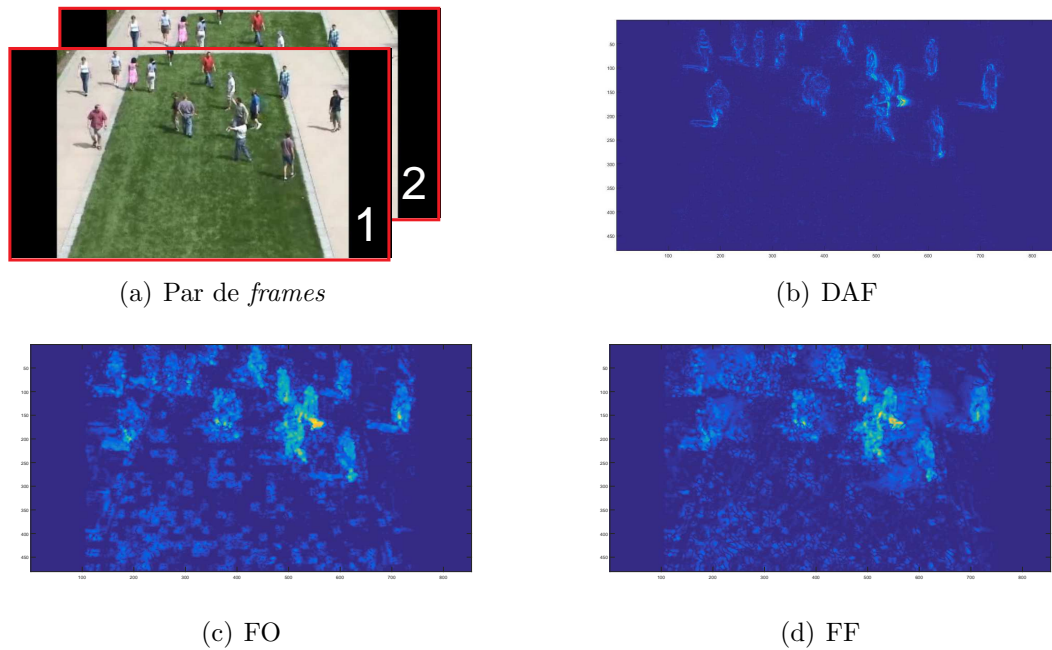


Figura 18: Em (a) é mostrado um par de *frames* consecutivos, (b) mostra a DAF do par de *frames*, (c) mostra o FO dos *frames* e (d) mostra o FF gerado pelo MFS a partir do FO mostrado em (c).

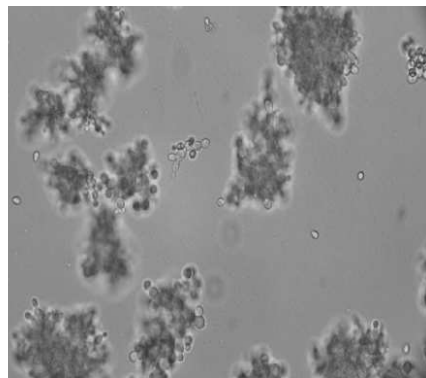
5.3 Colônia de Bactérias Artificiais

O algoritmo CBA desenvolvido nesta dissertação foi inspirado no algoritmo BFO e no comportamento natural das bactérias de se agruparem em colônias. As bactérias artificiais são pontos em um plano Cartesiano superposto sobre as camadas de movimento obtidas pela DAF, pelo FO ou pelo FF. Este plano possui as mesmas dimensões da camada. Cada bactéria avalia sua aptidão através de uma correspondência entre a sua posição no plano Cartesiano e a sua posição na camada a qual está superposta.

Como nas bactérias reais, as bactérias artificiais são regidas pelos princípios da seleção natural Darwiniana. Priorizam o forrageamento e a reprodução, mensuram o valor de sua posição correspondente na função de custo e estocam esse valor como se fosse alimento. A função de custo é a camada que contém o movimento extraído entre dois *frames* consecutivos. O valor da posição da bactéria artificial na função de custo revela sua aptidão para sobreviver ou reproduzir. A cada época evolutiva de uma camada, a bactéria avalia sua posição e armazena o valor de magnitude (aptidão) encontrado,

gerando um estoque de nutrientes ao longo das épocas. As bactérias que exploram regiões da função com baixos valores, tendem a gerar um estoque menor do que àquelas que estão em regiões com valores maiores. Assim, a camada de movimento torna-se um ambiente que estimula a competição entre as bactérias, e deste modo, sobrevivem apenas as que possuem aptidão acima do limiar de sobrevivência da camada e reproduzem apenas as que possuem um estoque de nutrientes acima do limiar de reprodução da mesma camada.

As Figuras 19(a) e (b) ilustram a analogia entre as bactérias reais e artificiais. As Figuras 19(c) e (d) exemplificam a implementação realizada nesta dissertação, em que as bactérias posicionam-se nas regiões que coincidem com a presença de quantidades elevadas de movimento entre *frames*.



(a) Colônia de bactérias reais



(b) Aglomerado de microrganismos em forma de mão



(c) *Frame* com pessoas em movimento



(d) Colônia de bactérias artificiais sobre as regiões com movimento

Figura 19: Em (a) é mostrado uma colônia de bactérias reais e em (b) as bactérias reais demarcando um ambiente nutricional na forma de mão. Em (c) mostra-se uma camada de movimento extraída de 2 *frames* consecutivos e em (d), mostra-se a distribuição das bactérias artificiais sobre essa camada.

A condição de sobrevivência (CS) define-se pela média e o desvio padrão das magnitudes de cada camada. A condição de reprodução (CR) define-se pela média e desvio padrão dos valores de estoque nutricional acumulado pelas bactérias sobreviventes. Assim,

os limiares são estimados dinamicamente em função das particularidades das camadas e, por isso, o algoritmo CBA adapta-se rapidamente as mudanças de cenário. A média e o desvio padrão da camada são calculados conforme a Equação 40:

$$\begin{aligned}\mu(k) &= \frac{1}{n} \sum_{j=1}^n C(x_j, y_j, k) \\ \sigma(k) &= \sqrt{\frac{1}{n} \sum_{j=1}^n [C(x_j, y_j, k) - \mu(k)]^2},\end{aligned}\tag{40}$$

onde $\mu(k)$ e $\sigma(k)$ são respectivamente a média e o desvio padrão das magnitudes da camada k , n é o número total de magnitudes e $C(x_j, y_j, k)$ é a camada de movimento, que pode ter sido gerada tanto pela DAF, quanto pelo FO ou FF.

A aptidão da bactéria b_i é mensurada através de uma correspondência entre a sua posição no plano Cartesiano e a posição da magnitude do movimento na camada conforme a Equação 41:

$$\begin{aligned}\text{Posição}(b_i) &= P(x_{bi}, y_{bi}) \\ \text{Aptidão}(b_i) &= C(x_{bi}, y_{bi}, k),\end{aligned}\tag{41}$$

onde $x_{bi} = \{ 1, 2, \dots, \text{número de colunas} \}$ e $y_{bi} = \{ 1, 2, \dots, \text{número de linhas} \}$. A aptidão da bactéria b_i está associada à camada de movimento $C(x_{bi}, y_{bi}, k)$ e acumula-se apenas nas bactérias que sobrevivem ao longo das épocas evolutivas de uma colônia. Ressalta-se aqui que para cada camada existe apenas uma única colônia, e que tal colônia atinge sua maturidade apenas na última época evolutiva. O estoque de aptidões, ou estoque de nutrientes, é calculado conforme a Equação 42

$$\text{estoque}(b_i) = \sum_{epoca=1}^{N_e} C(x_{bi}, y_{bi}, k),\tag{42}$$

onde N_e é o número de épocas evolutivas das colônias. A cada época o número de bactérias da colônia se altera devido a eliminação e reprodução. O estoque na última época consiste, deste modo, dos valores acumulados apenas pelas bactérias que sobreviveram à evolução. A média e o desvio padrão dos nutrientes acumulados pelas bactérias sobreviventes é calculado conforme a Equação 43:

$$\begin{aligned}\mu_{estoque}(k) &= \frac{1}{N_f} \sum_{i=1}^{N_f} estoque(b_i) \\ \sigma_{estoque}(k) &= \sqrt{\frac{1}{N_f} \sum_{i=1}^{N_f} [estoque(b_i) - \mu_{estoque}]^2},\end{aligned}\tag{43}$$

onde o $\mu_{estoque}(k)$ é a média do estoque das bactérias sobreviventes, N_f é o número de bactérias na época corrente, $estoque(b_i)$ é o estoque da bactéria (b_i) sobrevivente e $\sigma_{estoque}(k)$ é o desvio padrão do estoque nutricional da camada. Com isso a CS e a CR são então estimados conforme a Equação 44:

$$\begin{aligned}CS(k) &= \mu_{magnitudes}(k) + s * \sigma_{magnitudes}(k) \\ CR(k) &= \mu_{estoque}(k) + r * \sigma_{estoque}(k),\end{aligned}\tag{44}$$

onde s e r são os parâmetros do algoritmo CBA que definem a sensibilidade do processo de seleção natural. Variações no parâmetro s influenciam no espalhamento das bactérias sobre as regiões de interesse e variações em r influenciam na concentração de bactérias nessas regiões.

O centroide é o ponto cujas coordenadas são as médias das coordenadas dos pontos que formam uma figura geométrica, ou no caso desta aplicação, da forma geométrica resultante da distribuição das bactérias. O centroide da colônia e o módulo do centroide, são encontrados a partir da posição das bactérias sobreviventes na última época, conforme a Equação 45:

$$\begin{aligned}\text{centroide}(k) &= \left(\frac{1}{N_b} \sum_{i=1}^{N_b} x_b(i), \frac{1}{N_b} \sum_{i=1}^{N_b} y_b(i) \right) \\ \text{módulo do centroide}(k) &= \sqrt{\left(\frac{1}{N_b} \sum_{i=1}^{N_b} x_b(i) \right)^2 + \left(\frac{1}{N_b} \sum_{i=1}^{N_b} y_b(i) \right)^2},\end{aligned}\tag{45}$$

onde N_b é o número total de bactérias sobreviventes na última época, e $x_b(i)$ e $y_b(i)$ são as posições das bactérias no eixo horizontal e vertical, respectivamente.

O Algoritmo 6 apresenta os passos da CBA. Define-se em primeiro lugar, o tamanho da população e a posição inicial das bactérias na primeira época das camadas. Na primeira otimização, as bactérias da população inicial são distribuídas aleatoriamente, e nas otimizações seguintes essa mesma configuração se mantém. Assim, a convergência do algoritmo parte sempre de uma mesma configuração inicial da população de bactérias.

Define-se em seguida, o número de épocas para cada camada, levando em consideração que a cada época, a partir da segunda, o número de bactérias praticamente dobra. Este é um parâmetro que tem forte influência no esforço computacional exigido pelo algoritmo. Um número de épocas inferior a 3 cria instabilidades na população final das colônias e um valor superior a 6 não gera diferenças significativas nos resultados. Por isso, o valor definido nesta dissertação manteve-se em 5 durante todos os experimentos. Por último, define-se a condição de reprodução, que é um parâmetro de elitização para permitir que somente as melhores bactérias reproduzam. A condição de reprodução é responsável pela concentração elevada de bactérias nos pontos mais altos da função de custo, equivalente as regiões de maior movimento entre *frames*. Após as configurações dos parâmetros, existe um laço para otimizar as camadas, outro laço para as épocas evolutivas em cada camada e um último laço para executar o processo de seleção natural para cada bactéria de uma época. Por fim, o algoritmo fornece os registros de população, estoque nutricional e módulo do centroide de k colônias de bactérias.

Na etapa de eliminação, as bactérias da colônia que possuem uma aptidão inferior à CS, não estarão aptas para sobreviver, similar ao que aconteceria com bactérias em um ambiente real. Nesta etapa, principalmente nas primeiras épocas, ocorre a eliminação da maior parte das bactérias localizadas em regiões da camada com ruído ou baixas magnitudes de movimento entre *frames*. A partir de então, o processo reprodutivo ganha ênfase e há uma explosão populacional nas regiões com movimento intenso. Essa explosão populacional, cujo crescimento é exponencial, delimita o contorno da região.

Na etapa de reprodução, as bactérias sobreviventes localizadas nos ótimos locais da função de custo, se reproduzem assexuadamente, dividindo-se em duas idênticas. As duas bactérias são posicionadas aleatoriamente na vizinhança do ponto em que ocorreu a reprodução. Esse pequeno deslocamento que ocorre a cada época simula um passo quimiotático, que é diferente da quimiotaxia do algoritmo BFO, e não requer esforço computacional extra. Assim, a cada época existe um pequeno espalhamento de bactérias responsável por delimitar o contorno das regiões de interesse. As bactérias que se distanciam muito são eliminadas pela escassez de nutrientes.

A Tabela 1 realiza uma comparação entre os algoritmos BFO e CBA. A otimização por CBA possui a população de bactérias variável e dependente das regiões de interesse entre *frames*. O passo quimiotático ocorre logo após a reprodução, quando as duas bac-

Algoritmo 6 Algoritmo de colônia de bactérias artificiais (CBA)**Entrada:** volume espaço-temporal com k camadas de movimento;**início** Define N ;

Define a posição das bactérias na primeira época;

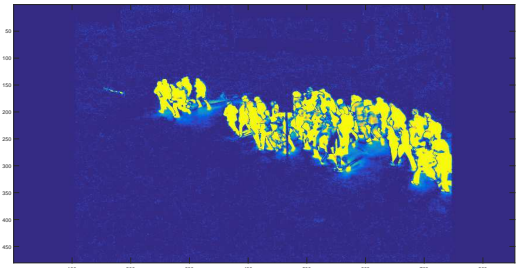
 Define N_e ; Define s ; Define r ; **para** cada camada k **faça** Calcule $\mu_{magnitudes}$ e $\sigma_{magnitudes}$; Calcule CS **para** cada época e **faça** **para** cada bactéria i **faça** Avalie aptidão da bactéria (i) **se** aptidão da bactéria (i) $\geq CS$ **então** estoque (i) := estoque (i) + aptidão da bactéria (i); **fim** **fim** **para** cada bactéria i **faça** **se** aptidão da bactéria (i) $\leq CS$ **então** elimine bactéria (i); **fim** **fim** Calcule $\mu_{estoque}$ e $\sigma_{estoque}$; Calcule CR **para** cada bactéria i **faça** **se** estoque(i) $> CR$ **então** bactéria (i) reproduz; **fim** **fim** **fim** calcule o módulo do centroide da colônia da camada k ; **fim****fim****Saída:** k populações; k estoques; k módulo dos centroides;

térias são reposicionadas aleatoriamente na vizinhança do ponto onde ocorreu a divisão. No CBA, as bactérias são entidades mais simples e não realizam giro ou nado. A ênfase do algoritmo é dada as características associadas à formação de colônias. O número de bactérias que sobrevivem e reproduzem está diretamente associado à média das magnitudes da camada de movimento e à média das aptidões acumuladas pelas bactérias sobreviventes, respectivamente. Isso introduz forte dinamismo à otimização e elevado grau de adaptabilidade aos mais diversos cenários.

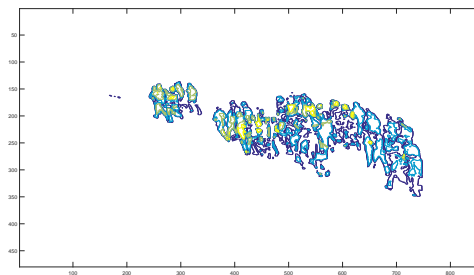
Tabela 1: BFO *vs.* CBA

	BFO	CBA
População	constante	variável
Passo quimiotático	Sim	Sim (ocorre na reprodução)
Giro	Sim	Não
Nado	Sim	Não
Formação de colônia	Sim	Sim
Reprodução	Condição fixa	Condição variável
Eliminação	Probabilística	Determinística
Dispersão	Sim	Não

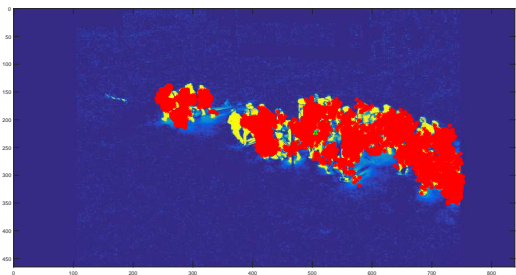
A Figura 20 ilustra a otimização de uma camada de movimento. Na Figura 20(b) é mostrado a camada de movimento correspondente ao *frame* apresentado em (a), com as regiões de movimento intenso em destaque. Em (c) coloca-se em evidência o contorno das regiões de interesse e em (d) é mostrado como a colônia de bactérias recobre as áreas de movimento após a otimização.

(a) *Frame*

(b) Camada de movimento



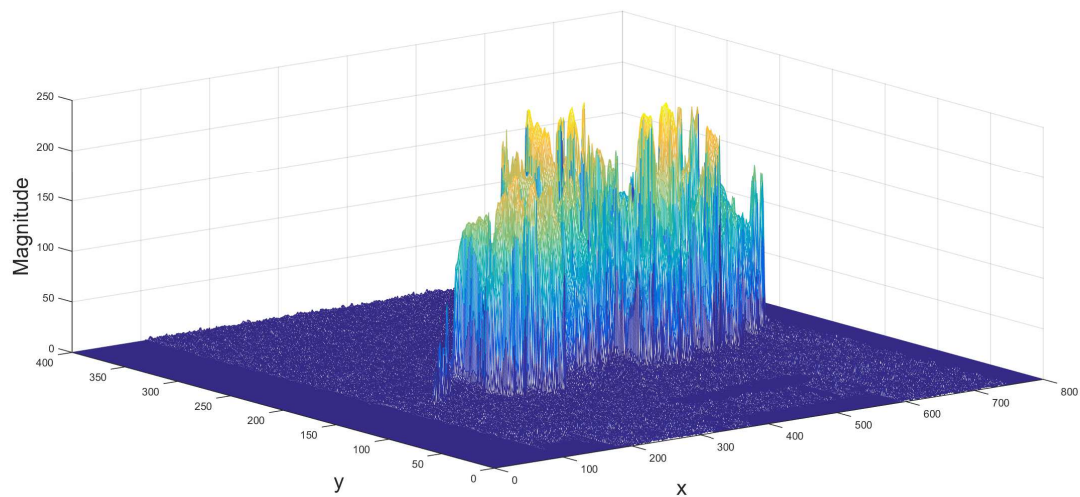
(c) Contorno das regiões com movimento



(d) Colônia de bactérias da camada

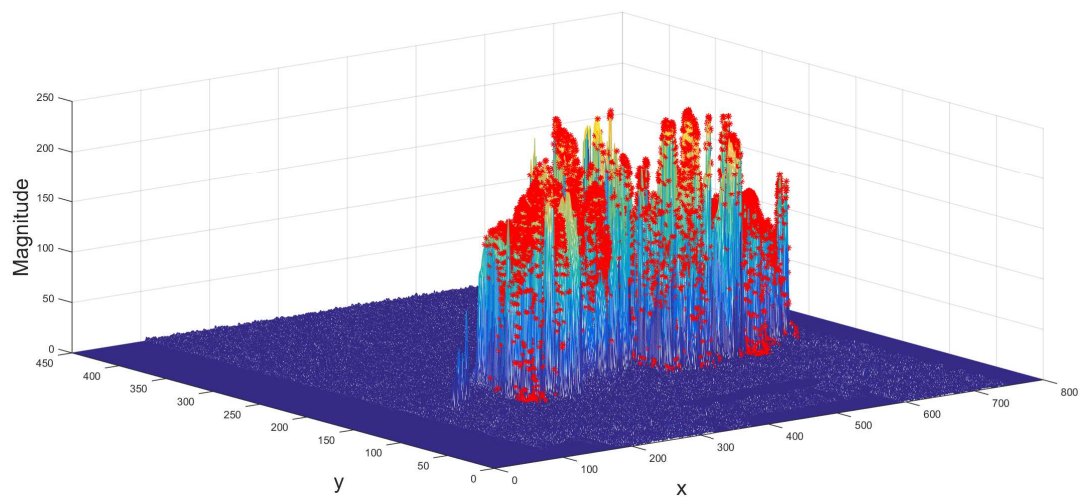
Figura 20: Ilustração da otimização de uma camada de movimento

As Figuras 21 e 22 ilustram como as bactérias sobrevivente se distribuem na a função de custo. É possível notar que o ruído está presente na função em magnitudes pouco atrativas para as bactérias.



(a) Função de custo (camada de movimento)

Figura 21: Função de custo corresponde à camada de movimento da Figura 20(b)

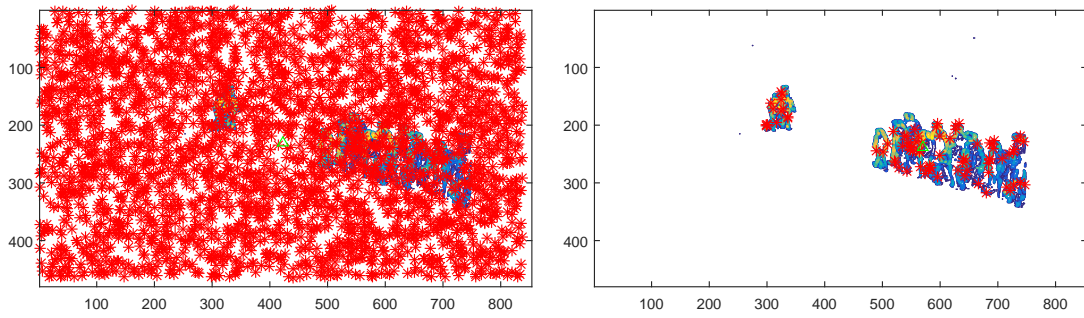


(a) Função de custo após a otimização

Figura 22: Função de custo da Figura 21 recoberta com as bactérias artificiais após a otimização. Essa função equivale à camada otimizada mostrada na Figura 20(d)

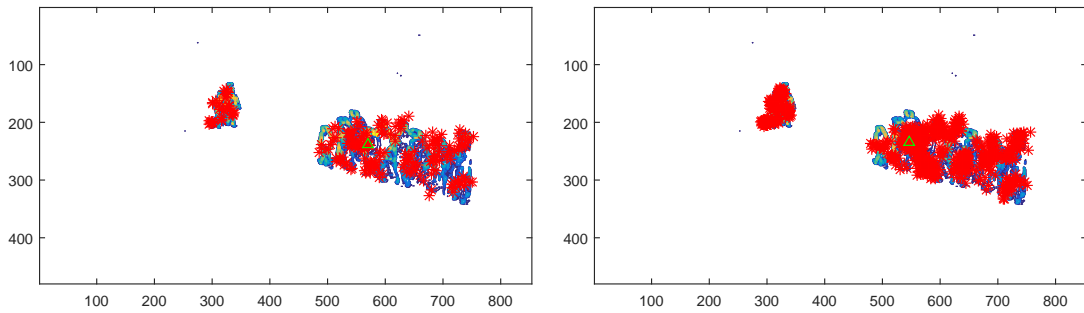
A evolução das bactérias ocorre ao longo de algumas épocas em uma camada, e o algoritmo converge para a formação da colônia nas áreas com as maiores magnitudes da função. A Figura 23 mostra a evolução da colônia de bactérias ao longo de 5 épocas da camada de movimento apresentada na Figura 20(b).

A população, o estoque de nutrientes e o módulo do centroide de cada colônia são registrados e utilizados posteriormente como conjunto de treinamento da RNK. Tais registros foram escolhidos devido às observações preliminares durante os primeiros experimentos, de que nas cenas em que os pedestres se deslocam em grupos com velocidades



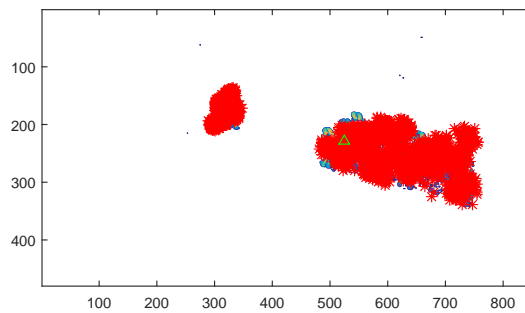
(a) Época 1

(b) Época 2



(c) Época 3

(d) Época 4



(e) Época 5

Figura 23: Evolução da colônia de bactérias ao longo de 5 épocas da camada de movimento apresentada na Figura 20(b). O pequeno triângulo verde em (e) representa a posição do centroide.

próximas, a população de bactérias tende a ser elevada, refletindo o tamanho da região associada ao movimento em cena. Em cenas em que apenas algumas pessoas correm, por exemplo, enquanto a maioria permanece caminhando, a população de bactérias varia

pouco, enquanto há um significativo aumento nas magnitudes da camada e consequentemente na qualidade de alimento disponível, gerando assim uma forte e rápida elevação no valor de nutrientes acumulados pelas bactérias. Por isso, o estoque nutricional reflete a quantidade de movimento entre *frames*. Já o centroide está fortemente associado à forma geométrica da colônia e reflete a direção para a qual os aglomerados de pessoas se deslocam na cena.

5.4 Redes Neural de Kohonen

A rede neural de Kohonen faz parte de um grupo de redes neurais que utilizam estratégias de competição para efetuar o ajuste de seus pesos no processo de aprendizado, são capazes de preservar as relações de vizinhança dos dados de entrada e utilizam treinamento não-supervisionado para encontrar similaridades com base apenas nesses padrões de entrada. Devido a tais características, a RNK foi escolhida para realizar a classificação de dados no sistema proposto nesta dissertação. Em (MARANA et al., 1998) uma aplicação semelhante foi implementada utilizando RNK para estimar a densidade de pessoas em multidões. E em (FENG; ZHANG; HAO, 2010) utiliza-se RNK para aprender e classificar em tempo real os padrões de *pixels* presentes em vídeos com tumultos na multidão.

As RNK aprendem a classificar os dados de entrada de acordo com a forma como estes são agrupados no espaço. Os neurônios da rede se tornam capazes de reconhecer seções vizinhas do espaço de entrada, por aprenderem a respeito da distribuição dos vetores. Neste trabalho, o conjunto de treinamento é formado por vetores contendo 3 informações sobre a colônia de bactérias de cada camada: o número de bactérias na última época, o estoque alimentar de todas as bactérias sobreviventes e o módulo do centroide da posição de todas as bactérias da colônia.

Nesta dissertação, o número de neurônios da rede foi escolhido empiricamente de modo a obter a melhor classificação dos eventos a o mesmo tempo exigir o menor custo computacional. Cada neurônio está associado a uma unidade de saída e assim representa uma respectiva classe. Durante a fase de treinamento, a rede neural de Kohonen determina a unidade de saída que melhor corresponde ao vetor de entrada e o vetor de pesos da unidade vencedora é ajustado de acordo com o algoritmo de treinamento competitivo. Durante o processo de auto-organização da rede neural, a classe cujo vetor de pesos mais se aproxima do vetor dos padrões de entrada é escolhida como sendo a vencedora. O

peso do neurônio vencedor é ajustado e os pesos dos neurônios vizinhos são atualizados segundo a função Gaussiana da distância do neurônio vencedor.

O algoritmo da RNK foi implementado utilizando a ferramenta SOM do MATLAB®, que permite, entre outras coisas, treinar a rede com diferentes topologias e parâmetros de aprendizagem, visualizar a distribuição dos usando matrizes e planos de componentes, codificação de cores em *cluster* etc. O número de neurônios, as dimensões da entrada e a estrutura do mapa devem ser especificados antes do treinamento. A atualização dos pesos da rede ocorrem em lotes segundo a utilização do algoritmo *batch training*, no qual os pesos sofrem a atualização somente após a apresentação de todas as amostras do conjunto de entrada. A ferramenta SOM apresenta um conjunto de informações relevantes a respeito da RNK, através dos seguintes funcionalidades, ilustradas na Figura 24:

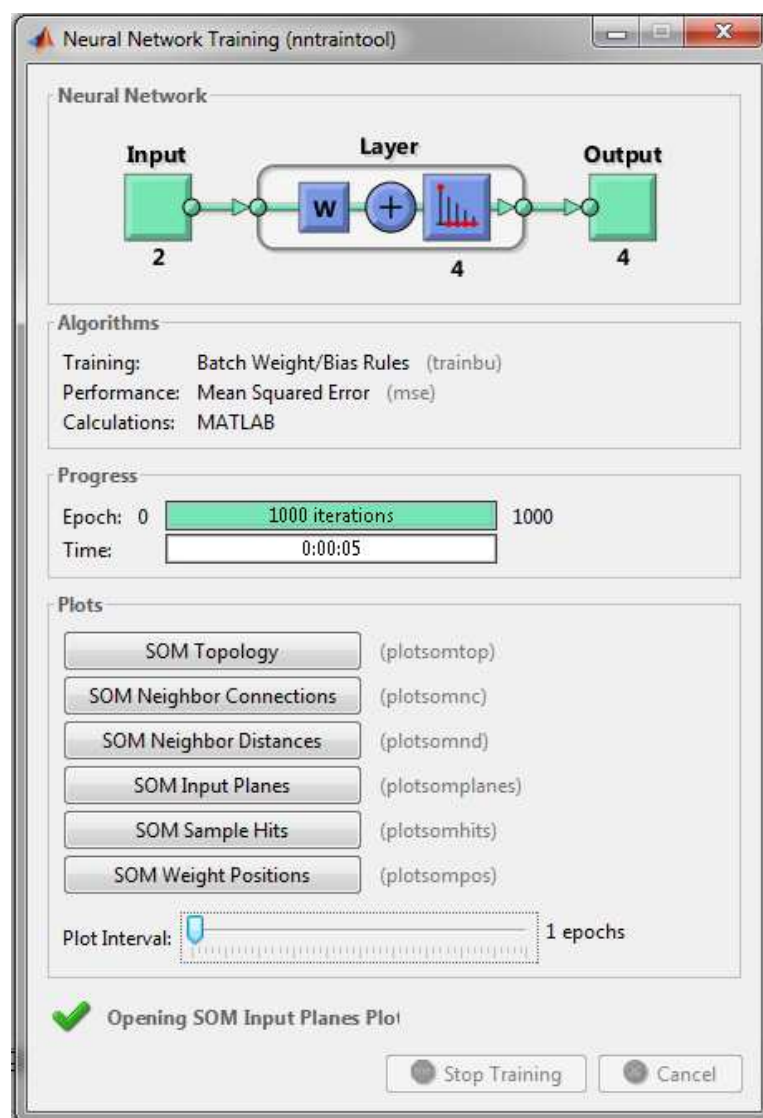


Figura 24: Janela de simulação da ferramenta SOM

- *SOM Topology*: permite visualizar a configuração topológica da rede.
- *SOM Neighbor Connections*: permite visualizar os neurônios e suas relações de vizinhança direta.
- *SOM Neighbor Distances*: permite visualizar o quão próximo estão neurônio de seus vizinhos.
- *SOM Input Planes*: mostra os pesos da i -ésima entrada para os neurônios da rede, com as conexões negativas em amarelo, as conexões próximas de zero em preto e as conexões mais positivas em vermelho.
- *SOM Sample Hits*: permite visualizar o número de vetores de entrada que cada neurônio está associado.
- *SOM Weight Positions*: permite visualizar a posição final dos pesos da rede após o treinamento.

Durante o treinamento da RNK, o vetor de peso associado a cada neurônio se move para posicionar-se no centro de um grupo de vetores de entrada que apresenta similaridades. Graças as propriedades da rede, é possível visualizar um espaço de entradas de alta dimensionalidade em 2 dimensões.

Os neurônios da rede são dispostos originalmente em posições aleatórias, de acordo com a topologia escolhida. As funções disponíveis são: *gridtop*, que organiza os neurônios em uma topologia de grade retangular no início do treinamento, *hexdtop*, que organiza os neurônios em um padrão hexagonal, e *randtop*, que realiza a organização em padrões aleatórios. Uma ilustração das topologias é mostrada na Figura 25.

A distância entre os neurônios pode ser calculada a partir de suas posições pelas funções de distância oferecidas pela ferrameta MATLAB: *dist*, que calcula a distância Euclidiana entre o neurônio vencedor e os neurônios vizinhos e esse neurônio vencedor, *linkdist* que obtém a distância entre um neurônio e os seus vizinhos com base apenas no número de links entre estes e *mandist*, que efetua o cálculo da distância entre o neurônio vencedor e o vizinho pela soma da diferença absoluta das respectivas coordenadas. A função *dist* foi utilizada neste trabalho para calcular a distâncias entre os neurônios e definiu-se a topologia *gridtop* para a rede. A Figura 26 ilustra o treinamento realizado pela RNK e resalta a posição inicial dos pesos, antes do treinamento Figura 26(a), e a

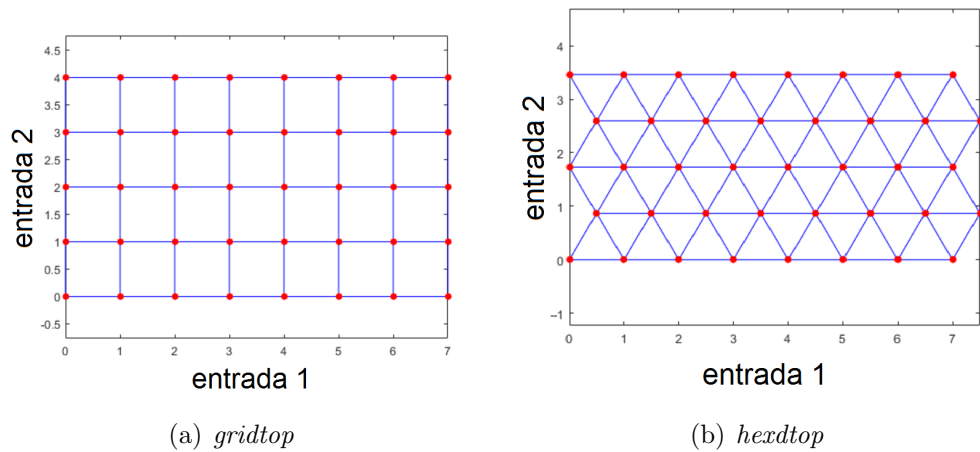
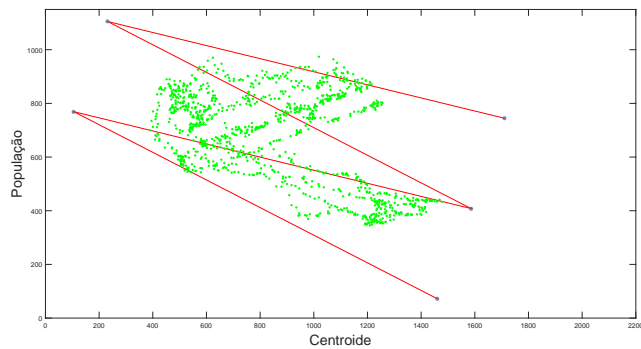
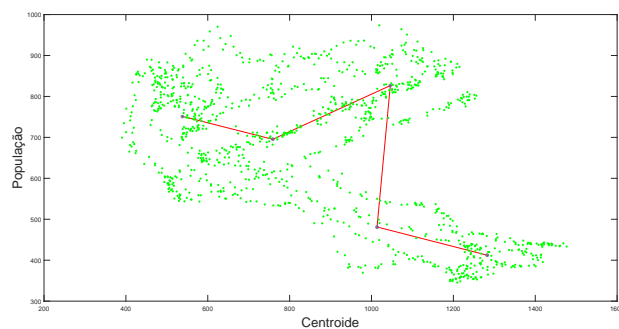


Figura 25: Topologias

posição final após o treino Figura 26(b). As RNK utilizadas nos experimentos possuem todas uma única camada apenas, e deste modo, a topologia *gridtop* apresenta-se como na Figura 26(a).



(a) Posição inicial dos pesos



(b) Posição final dos pesos

Figura 26: Posição dos neurônios antes e depois do treinamento

5.5 Considerações Finais

Este capítulo apresentou os principais aspectos de implementação do sistema proposto nesta dissertação. A extração de movimento utilizou as técnicas de DAF, FO e FF. O algoritmo CBA foi apresentado em detalhes e sua relação com o BFO foi evidenciada na Tabela 1. A RNK foi escolhida para efetuar o agrupamento dos registos comportamentais oriundos da otimização do algoritmo CBA e os aspectos de implementação foram tratados levando em consideração o uso da ferramenta SOM do MATLAB®.

Capítulo 6

EXPERIMENTOS E RESULTADOS

ESTE capítulo apresenta os resultados obtidos na implementação do sistema proposto no Capítulo 5. Para a extração de movimento, foram utilizadas as técnicas de DAF, FO e FF apresentadas no Capítulo 1. O algoritmo CBA, visto nos Capítulos 4 e 5 otimiza as camadas de movimento. A RNK descrita no Capítulo 4 classifica os registros da evolução das colônias de bactérias na etapa de otimização. Para a análise dos resultados foram utilizados: o tempo de pré-processamento por *frame* na extração de movimento, o tempo de otimização do algoritmo CBA por camada de movimento e a curva ROC da detecção dos eventos. A análise dos resultados permite avaliar a eficácia do algoritmo em realizar a detecção de eventos e a sua viabilidade para aplicações em tempo real.

A Seção 6.1 indica as métricas utilizadas. A Seção 6.2 mostra os parâmetros de configuração da CBA e da RNK. A Seção 6.3 apresenta os resultados obtidos nos experimentos com os vídeos do *dataset* PETS 2009. A Seção 6.4 expõe os resultados obtidos pela análise do *dataset* UMN. A Seção 6.5 mostra os resultados obtidos pela análise de 1 vídeo de longa duração gravado especificamente para avaliar a robustez do sistema proposto. Por fim, a Seção 6.6 apresenta as considerações finais.

6.1 Métricas Utilizadas

Para avaliar a qualidade dos resultados obtidos nos experimentos foram utilizadas as seguintes métricas de desempenho:

- Tempo médio de processamento por par de *frame* na etapa de geração de camadas de movimento.

- O *Speedup* para mensurar o ganho de tempo entre as técnicas de DAF, FO e FF na etapa de geração das camadas.
- Tempo médio de otimização do algoritmo CBA por camada.
- Tempo de treinamento da RNK.
- Curva ROC de todos os *frames* do *dataset* para avaliar a qualidade do classificador.

A lei de Amdahl define *speedup* (S) como o ganho em desempenho que pode ser obtido ao melhorar uma determinada tarefa computacional, conforme Equação 46:

$$Speedup(S) = \frac{\text{tempo de execução da operação sem usar a melhoria } M}{\text{tempo de execução da operação usando a melhoria } M} \quad (46)$$

É importante ressaltar que o *speedup* é obtido a partir da taxa de pré-processamento por *frame* na extração de movimento pela DAF, em relação ao FO e FF.

A curva de características de operação do receptor, ou curva ROC (*Receiver Operating Characteristic*), como também é conhecida, trata-se de uma métrica utilizada para verificar qualitativamente as propriedades de detecção do classificador.

Os gráficos ROC foram recentemente introduzidos como uma poderosa e simples ferramenta para a avaliação de algoritmos de aprendizado. A curva ROC plota as medidas de sensibilidade versus especificidade para todos os possíveis pontos de corte entre 0 e 1.

A sensibilidade, que está associada ao eixo vertical, é a proporção entre os resultados verdadeiros positivos, ou seja, essa métrica que avalia a capacidade efetiva do modelo para classificar um indivíduo como evento $Y = 1$ dado que em verdade, ele realmente é o evento $Y = 1$:

$$Sensibilidade = \frac{VP}{VP + FN}, \quad (47)$$

onde VP é um verdadeiro positivo e FN é um falso negativo. Já a especificidade, associada ao eixo horizontal, refere-se à proporção de verdadeiros negativos, isto é, a capacidade do modelo em prever um indivíduo como não sendo o evento $Y = 0$ dado que ele realmente não é o evento $Y = 0$, onde VN é um verdadeiro negativo e FP é um falso positivo.

$$Especificidade = \frac{VN}{VN + FP}, \quad (48)$$

6.2 Configurações do Algoritmo

Para avaliar o desempenho do método proposto foram realizados extensivos experimentos nos *datasets* públicos UMN (UMN, 2006) e PETS 2009 (PETS, 2009), convencionalmente utilizados em análises de sistemas para detecção de eventos em multidão. E para verificar a robustez do método, experimentos também foram realizados em 1 vídeo de longa duração filmado especificamente para este trabalho.

Todos os vídeos dos *datasets* PETS 2009 e UMN possuem resolução de 480×854 *pixels*. Todos os algoritmos foram executados com o Matlab R2015a. Todas as simulações foram realizadas em PC com Intel Core i7 950 3GHz, 8Gb RAM e sistema de operação Microsoft Windows 7 Home Premium.

Nos experimentos, durante a etapa de otimização do sistema proposto, as populações de bactérias são aleatoriamente espalhadas na primeira época de cada camada de movimento. O número de bactérias da população inicial variou de 250, 500, 1000 a 2000, para verificar as variações no tempo de otimização e na diferença entre as curvas de populações finais. Os outros parâmetros do algoritmo foram fixados para todas as simulações da seguinte forma:

- $N_e = 5$
- $s = 1, 5$
- $r = 2$

onde o valor do número de épocas N_e mostrou-se ser, em geral, o mais vantajoso computacionalmente e qualitativamente. O parâmetro da condição de sobrevivência s influencia na sobrevivência das bactérias cuja aptidão está acima de CS e o parâmetro de reprodução r reflete na divisão das bactérias cujo estoque de aptidão acumulado está acima de CR .

Para treinar a RNK foi utilizado o algoritmo padrão *batch weight* que realiza o treinamento a partir de lotes de dados na entrada. O número de épocas de treinamento foi fixado em 1000 porque esse valor mostrou-se satisfatório para agrupar confiavelmente a quantidade de dados fornecidos à rede neste trabalho, incluindo os dados provenientes do vídeo de longa duração. A quantidade de neurônios foi definida para cada vídeo como sendo proporcional ao número de eventos observados no vídeo a partir do ponto

de vista humano. Ou seja, se um o grupo de pessoas apresentarem os comportamentos de caminhar, parar de repente e em seguida fugir em disparada, definir-se-ia 3 neurônios para a rede responsável por esse vídeo. Cada vídeo possui a sua quantidade específica de neurônios que não se alterou ao longo dos experimentos.

6.3 *Dataset* PETS 2009

Essa seção descreve os resultados obtidos nos experimentos realizados com o *dataset* PETS 2009. Neste *dataset* são considerados anormais os eventos em que as pessoas comecem a correr ou se aglomerar repentinamente. Tais eventos se iniciam de forma suave, com uma transição gradual do comportamento normal para o comportamento anormal. Esse aspecto presente em todo o *dataset* torna a detecção uma tarefa desafiadora. A Figura 27 mostra *frames* representando cada vídeo do *dataset* PETS 2009.

(a) Video 1HL - 670 *frames*(b) Video 2HL - 1002 *frames*(c) Video 3HL - 394 *frames*(d) Video 4HL - 1134 *frames*(e) Video 1MF - 322 *frames*(f) Video 2MF - 697 *frames*(g) Video 3MF - 328 *frames*(h) Video 4MF - 511 *frames*(i) Video 5MF - 280 *frames*

Figura 27: *Frames* representando os vídeos do *dataset* PETS2009

No primeiro estágio do sistema, é realizado o pré-processamento dos vídeos para obter as camadas de movimento. Essa fase é importante, porque está associada a qualidade das camadas geradas e em geral exige bastante esforço computacional devido a quantidade de cálculos matemáticos. Para analisar o intervalo médio de tempo necessário para gerar uma única camada a partir de dois *frames* consecutivos, experimentou-se as técnicas de DAF, FO e FF. Os resultados obtidos são apresentados na Figura 28. Nota-se a necessidade do uso de uma escala logarítmica devido a distância existente entre o valor do tempo médio de processamento para a DAF em relação às técnicas de FO e FF. É válido ressaltar também que o tempo exigido pelo FF sempre será maior que o FO, porque a aplicação do MFS para obter o FF ocorre a partir das camadas de FO. Outro ponto importante é o tempo variável de pré-processamento no uso do FO e conseqüentemente do FF, porque o tamanho do vídeo influencia diretamente na quantidade de cálculos realizados por essas técnicas.

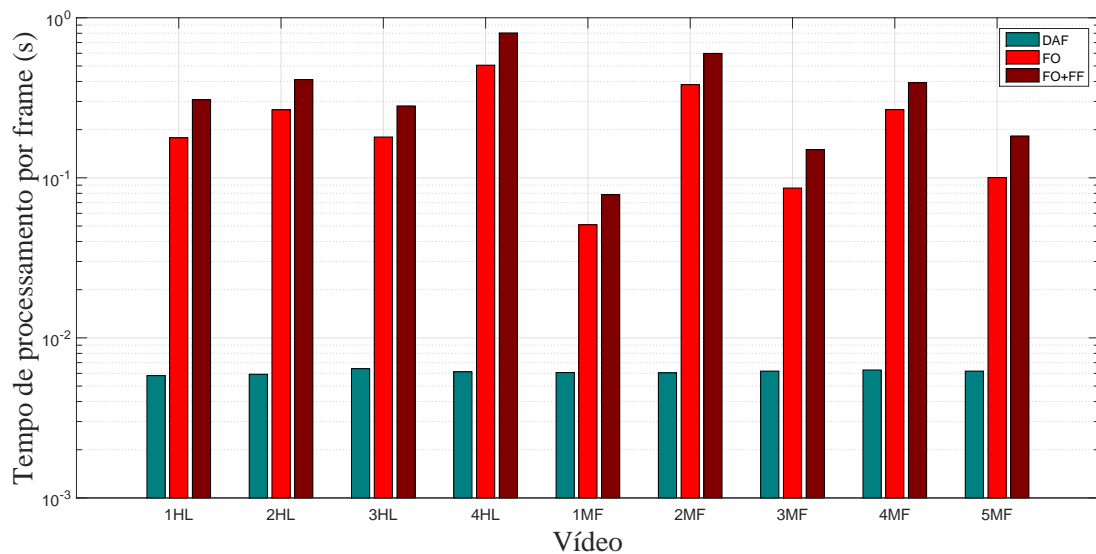


Figura 28: Tempo médio de processamento para geração de uma camada de movimento a partir de dois *frames*.

A Tabela 2 mostra os valores de *speedup* encontrados para ganho no tempo de pré-processamento pelo uso da DAF em relação ao FO e ao FF.

Para avaliar o tempo requerido pelo algoritmo CBA na otimização das camadas de movimento, foram realizados experimentos com populações iniciais de 250, 500, 1000 e 2000 bactérias. As Figuras 29, 30 e 31 apresentam os valores de tempo médio de otimização para camadas obtidas pela DAF, FO e FF. Os resultados mostraram que populações iniciais de 250 e 500 bactérias permitem uma otimização em intervalos que

Tabela 2: *Speedup* para o dataset PETS2009

Vídeo	FO para DAF	FF para DAF
1HL	30,61	53,02
2HL	44,88	69,33
3HL	28,01	43,82
4HL	82,38	131,01
1MF	8,38	12,92
2MF	63,21	99,09
3MF	13,92	24,17
4MF	42,35	62,26
5MF	16,18	29,45

são, em média, inferiores a 0,033 segundos. Esse desempenho foi observado para todas as técnicas de extração de movimento, revelando assim que para o algoritmo CBA, a procedência da camada de movimento não influencia diretamente no tempo médio de otimização. Pode-se também, partindo destes resultados, verificar um indício de que tais parâmetros possibilitariam a otimização em intervalos de tempo inferiores a taxa de *frame* dos vídeos analisados, que é de 30 frames/segundo.

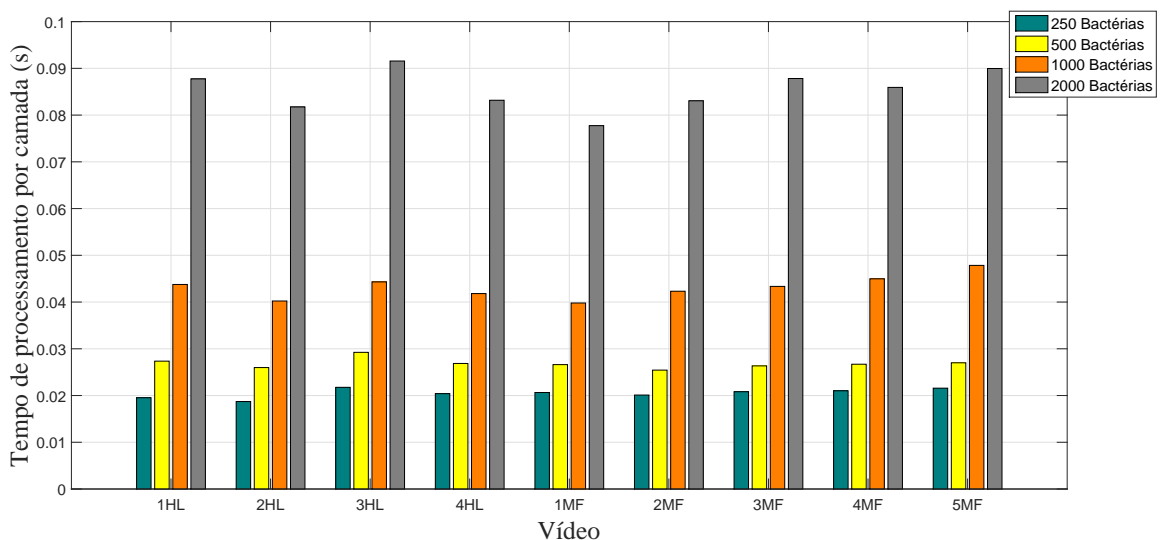


Figura 29: Tempo médio de otimização por camada de movimento obtido pelo FO.

Naturalmente é preciso que se some ao tempo médio de otimização, o tempo requerido para a geração da camada no estágio anterior. A Figura 28 revela que o uso da DAF permite gerar as camadas com uma média de 0,006 segundos/*frame*. Assim, fica desde já a ressalva de que utilizando a DAF no primeiro estágio e 250 ou 500 bactérias no segundo estágio, seriam obtidos respectivamente tempos médios de aproximadamente 0,026 segundos/*frame* e 0,031 segundos/*frame*. Ambos ainda dentro da janela da taxa

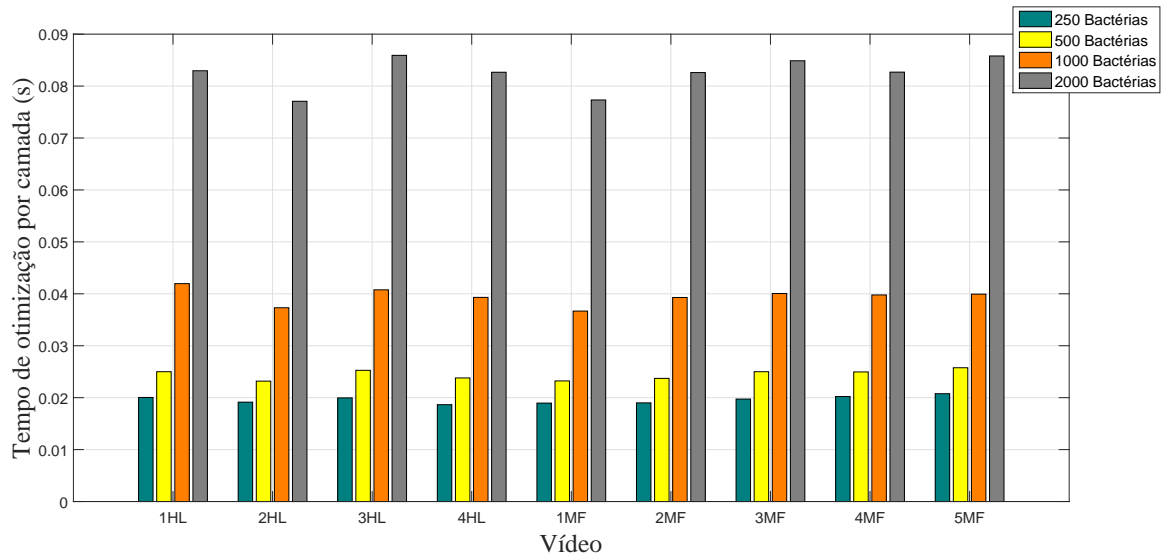


Figura 30: Tempo médio de otimização por camada de movimento obtido pelo FF.

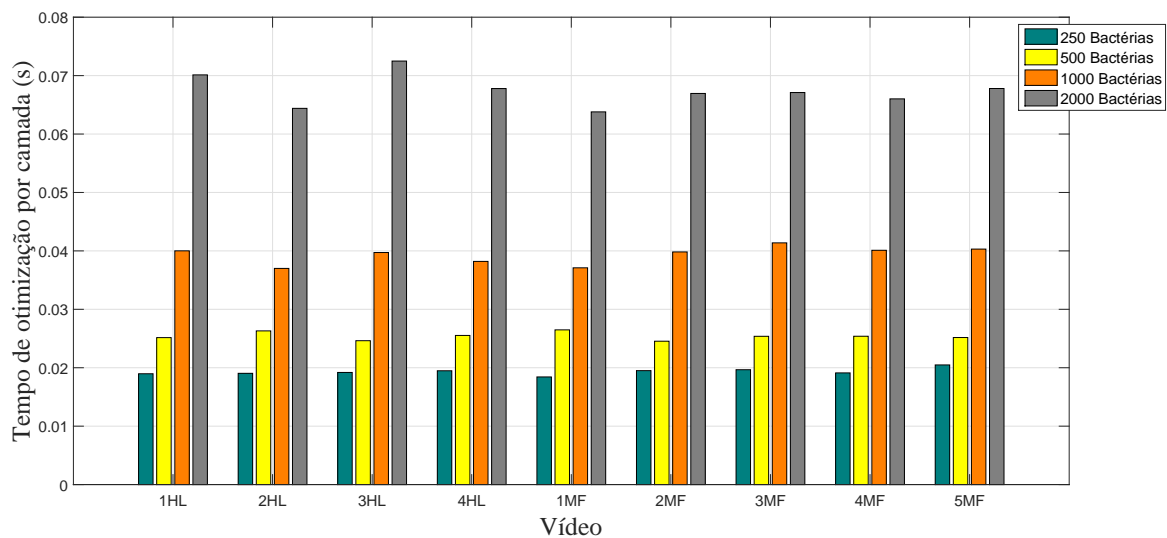


Figura 31: Tempo médio de otimização por camada de movimento obtido pela DAF.

de *frames* do vídeo.

No entanto, é preciso averiguar também a questão da perda de precisão do algoritmo pela redução do número de bactérias. A Figura 32 obtida sob a configuração de 250 bactérias iniciais e a Figura 33 sob a configuração 500, revelam o histórico das colônias geradas ao longo do vídeo 1HL. Este gráfico permite, inclusive, visualizar com clareza a forma como as colônias de bactérias espelham o comportamento da multidão. No vídeo 1HL, as pessoas começam a caminhar da direita para a esquerda e, repentinamente, começam a correr. Posteriormente, retornam da esquerda para a direita e mais uma vez iniciam o comportamento de fuga. Nos dois momentos em que se inicia a corrida, os valores das populações e estoque se elevam significativamente acima da média. Já o centroide revela em que posição da tela se encontra o núcleo da multidão. Como as pessoas foram para um

canto do vídeo e depois retornam, o centroide dos últimos *frame* tem um valor próximo ao dos primeiros. Nota-se, pela observação dos gráficos, que embora haja certa perda de precisão para 250 bactérias na população inicial, a forma de onda permanece praticamente inalterada. O mesmo ocorre para populações de 1000 e 2000 bactérias. Ressalta-se aqui também o fato de que nas Figuras 32 e 33 as camadas de movimento foram geradas pela DAF. A utilização da nomenclatura *frame* no eixo horizontal ocorre por uma questão de praticidade, já que sendo a camada correspondente aos pares consecutivos de *frames*, a posição da camada k coincide com a posição do *frame* f , exceto para o último.

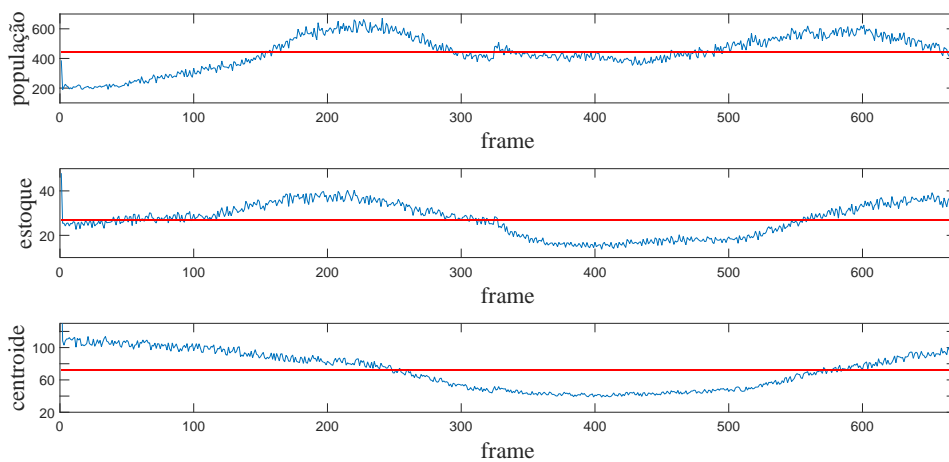


Figura 32: Registro das populações, estoques e centroides das colônias de bactérias obtidas ao longo da otimização do vídeo *1HL*, para 250 bactérias iniciais.

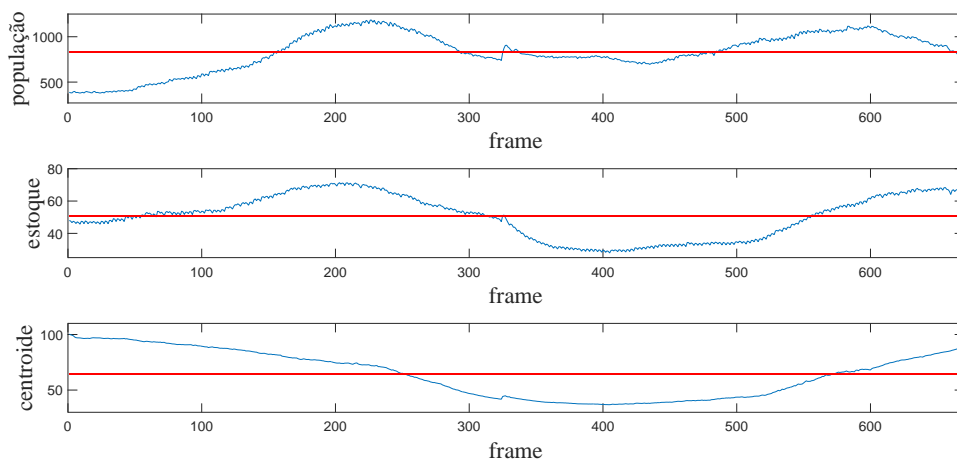


Figura 33: Registro das populações, estoques e centroides das colônias de bactérias obtidas ao longo da otimização do vídeo *1HL*, para 500 bactérias iniciais.

Para esse mesmo vídeo *1HL* tem-se três eventos bem definidos visualmente: pessoas caminhando, pessoas correndo em fuga da direita para esquerda e pessoas correndo

em fuga da esquerda para direita. Por isso, foram escolhidos 3 neurônios para agrupar os dados. A Figura 34 mostra a posição final dos pesos após o treinamento dos mesmos dados apresentados na Figura 32, contendo 250 bactérias na população inicial. Como o mapa com a posição dos pesos só pode ser gerado em duas dimensões, foram utilizados na ilustração o estoque e a população. A medida que o número de bactérias iniciais for maior, como 500, 1000 ou 2000, por exemplo, obteríamos uma concentração maior de pontos nas proximidades dos neurônios. A classificação, no entanto, ocorreria de forma muito semelhante.

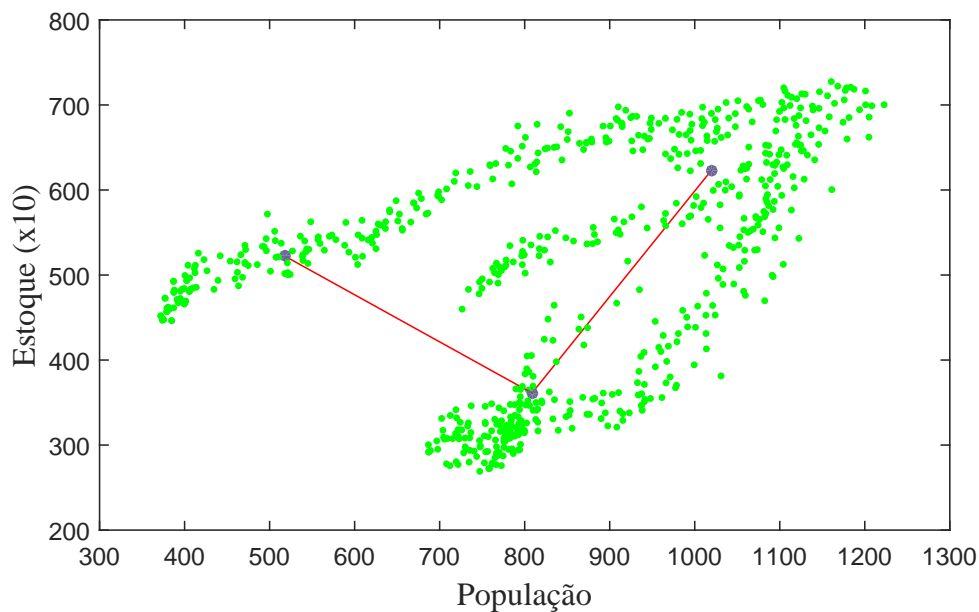


Figura 34: Distribuição dos pesos da RNK após o treinamento.

O número de neurônios definido para cada vídeo do *dataset* é apresentado na segunda coluna da Tabela 3. Na terceira coluna da tabela, é mostrado também o tempo médio de treinamento das redes de todos os vídeos, para otimizações com 250, 500, 1000 e 2000 bactérias iniciais, levando em consideração as 3 técnicas de extração de movimento. A quarta coluna apresenta o desvio padrão do tempo de treinamento, revelando que em geral, o treino para os vídeos do *dataset* PETS 2009 durou aproximadamente 5 segundos. A última coluna da tabela revela o tempo de processamento médio para a classificação de apenas uma nova amostra de dados após o treinamento da RNK. Os valores obtidos estão próximos de 0,000003 segundos e revelam principalmente que a rede neural de Kohonen, quando está devidamente treinada, é capaz de realizar a classificação muito rapidamente de novas amostras provenientes da posterior otimização.

É importante salientar que devido a curta duração dos vídeos e das especificidades da RNK, o próprio conjunto de treinamento foi utilizado para avaliar o classificador. O intuito foi observar a capacidade da rede de classificar os eventos já ocorridos no vídeo. Na seção de trabalhos futuros comenta-se a possibilidade de averiguar as potencialidades do sistema proposto para predição de anormalidades em vídeos novos do mesmo cenário.

Tabela 3: Número de neurônios, tempo médio e desvio padrão do treinamento, e tempo de classificação após o treino para cada vídeo do *dataset* PETS2009

Vídeo	N ^o de neurônios	Treinamento (s)	Desvio padrão	Amostra (s)
1HL	3	4,1625	0,5720	3,5366e-06
2HL	4	5,4535	0,6136	3.8084e-06
3HL	4	3,3218	0,4120	3.5661e-06
4HL	4	5,5329	0,3981	2.4690e-06
1MF	4	2,1832	0,4678	4.9582e-06
2MF	3	4,2577	0,5235	3.7385e-06
3MF	4	3,3579	0,3719	3.7673e-06
4MF	3	3,5885	0,2258	1.0873e-06
5MF	3	3,0601	0,4564	2.1352e-06

A detecção dos eventos está em concordância com a observação humana e com trabalhos anteriores como (RAGHAVENDRA et al., 2011; CHEN; HUANG, 2011; RAO et al., 2013), que foram utilizados como referência para validar a classificação da RNK. A Figura 35 ilustra a comparação da detecção realizada pelo método proposto com a detecção esperada. Para ilustrar as classes neste exemplo da Figura 35, utilizou-se a DAF para a extração de movimento e 500 bactérias na otimização. A diversidade no número de classes se dá pela variação do número de neurônios em cada vídeo. A classe de anormalidade foi normalizada como sendo igual a 1 e as classes restantes entre 0 e 1.

Para avaliar a taxa de detecções verdadeiras e falsas, a Figura 36 apresenta a curva ROC de todos os frames do *dataset* PETS 2009, para os 3 métodos de extração de movimento. A área sob a curva ROC (AUC) (*Area Under ROC Curve*) é utilizada como métrica para comparar a sensibilidade e especificidade da classificação para a DAF, FO e FF. Nota-se que os valores estão muito próximos, com uma pequena vantagem para FO.

A Tabela 4 expõe os resultados obtidos neste trabalho em contraste com a AUC obtida por (MEHRAN; OYAMA; SHAH, 2009), que utiliza apenas o FO para extrair movimento dos *frames* e grupo de palavras para realizar a classificação. O método proposto apresentou valores AUC superiores ao estado da arte do método referenciado.

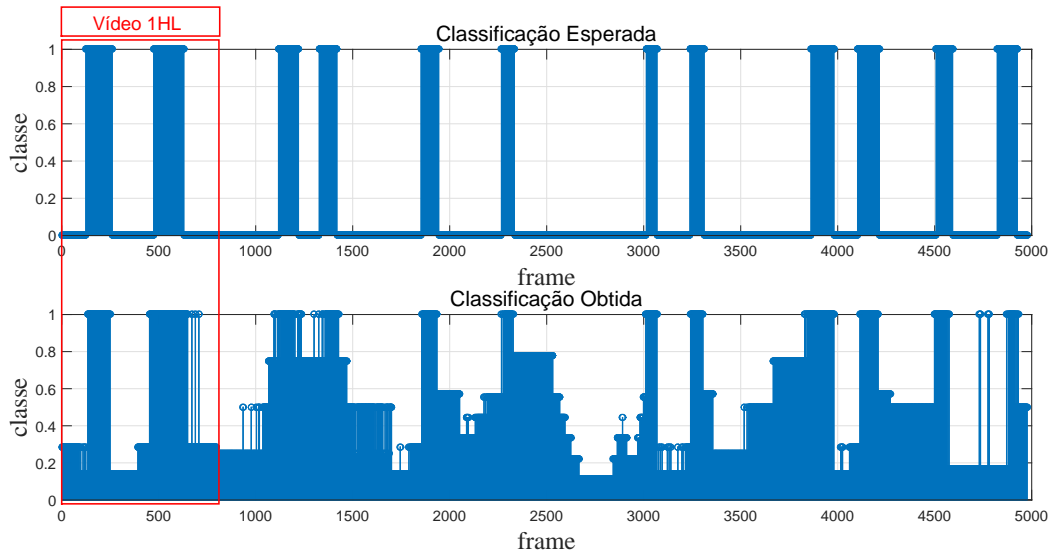


Figura 35: Comparação entre as classes esperadas, que possuem um evento considerado anormal, e as classes obtidas pelo método proposto para todos os frames do *dataset* PETS2009.

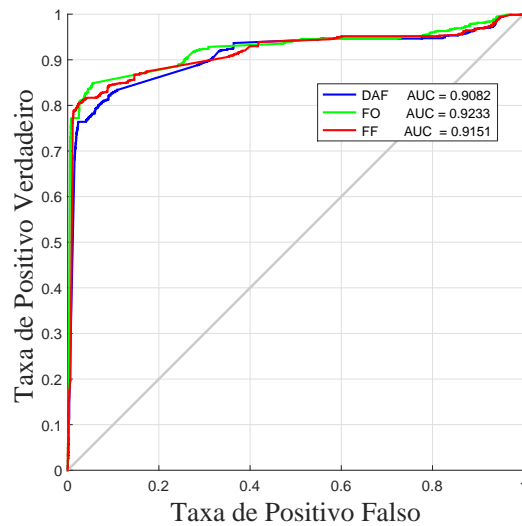


Figura 36: Curva ROC das detecções realizadas em todos os vídeos do *dataset* PETS2009 utilizando as 3 diferentes técnicas de extração de movimento.

Tabela 4: Área sob a curva ROC do método proposto *vs.* outros métodos, para o *dataset* PETS 2009

Método	Área sob a curva ROC	Melhoramento
FO (MEHRAN; OYAMA; SHAH, 2009)	0.8400	–
CBA - FO	0.9233	9,91%
CBA - FF	0.9151	8,93%
CBA - DAF	0.9082	8,11%

6.4 Dataset UMN

O *dataset* UMN (UMN, 2006) consiste de onze vídeos associados a três cenários diferentes contendo multidões de baixa e média densidade em ambientes fechados e abertos. Todas

as sequências de vídeos exibem situações de pânico entre os indivíduos. Os vídeos iniciam sempre com pedestres caminhando, em um comportamento convencionalmente classificado como normal, seguido de repentinamente tumulto e dispersão, caracterizando e atividade anormal. A Figura 37 mostra *frames* representando cada vídeo do *dataset*.

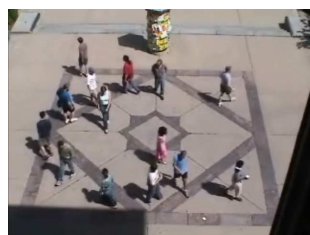
(a) Video 1 - 468 *frames*(b) Video 2 - 765 *frames*(c) Video 3 - 433 *frames*(d) Video 4 - 622 *frames*(e) Video 5 - 710 *frames*(f) Video 6 - 526 *frames*(g) Video 7 - 836 *frames*(h) Video 8 - 618 *frames*(i) Video 9 - 612 *frames*(j) Video 10 - 636 *frames*(k) Video 11 - 795 *frames*

Figura 37: *Frames* representando os vídeos do *dataset* UMN

No primeiro estágio do sistema é realizado o pré-processamento dos vídeos para obter as camadas de movimento. Os tempos médios de pré-processamento requerido para geração das camadas pelas técnicas de DAF, FO e FF são apresentados na Figura 38.

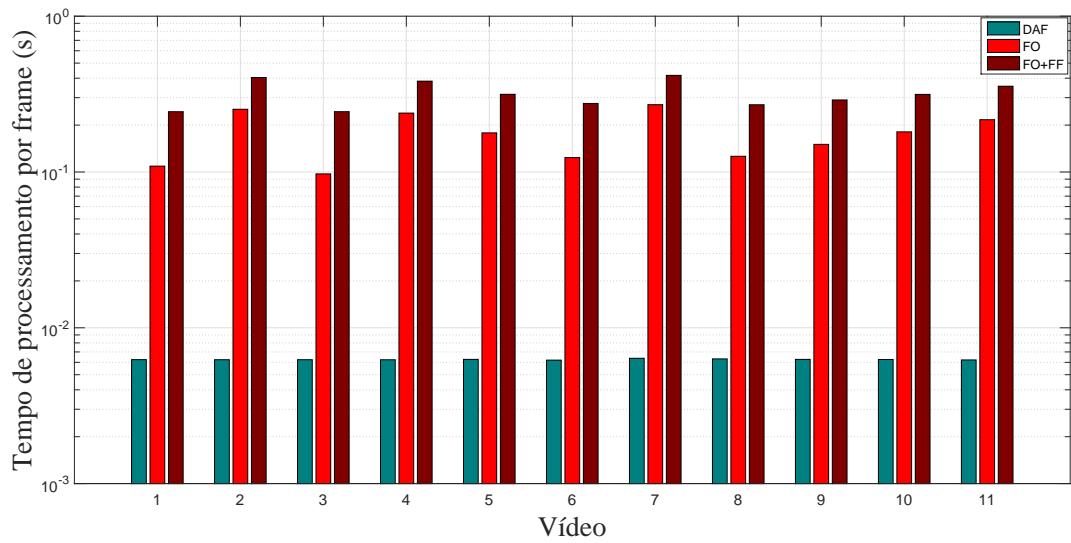


Figura 38: Tempo médio de processamento para geração de uma camada de movimento a partir de dois *frames* consecutivos.

A Tabela 5 mostra os valores de *speedup* encontrados para o ganho no tempo de pré-processamento pelo uso da DAF em relação ao FO e ao FF.

Tabela 5: *Speedup* no pré-processamento dos vídeos do *dataset* UMN

Vídeo	FO para DAF	FF para DAF
1	17,48	39,03
2	40,49	64,71
3	15,57	39,08
4	38,28	61,37
5	28,43	50,26
6	19,94	44,41
7	42,93	65,32
8	19,79	42,79
9	23,99	46,27
10	28,91	50,30
11	34,82	57,11

Novamente foram realizados experimentos com populações iniciais de 250, 500, 1000 e 2000 bactérias. As Figuras 39, 40 e 41 apresentam os valores de tempo médio de otimização para cada camada obtidas pela DAF, FO e FF. Os resultados mostraram que também para o *dataset* UMN os valores de 250 e 500 bactérias permitem uma otimização em intervalos que são, em média, inferiores a 0,033 segundos. Esse desempenho foi observado para todas as técnicas de extração, revelando mais uma vez que para o algoritmo CBA, a procedência da camada de movimento não influencia diretamente no tempo médio de otimização.

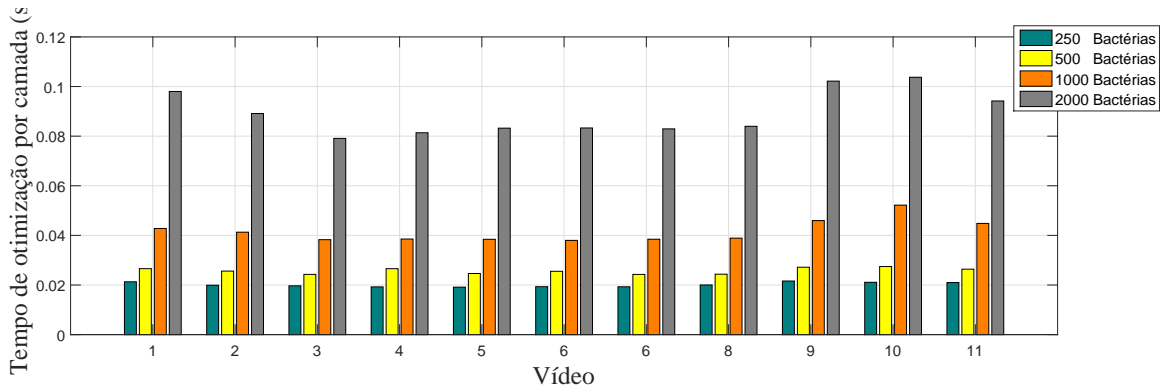


Figura 39: Tempo médio de otimização por camada de movimento obtido pelo FO.

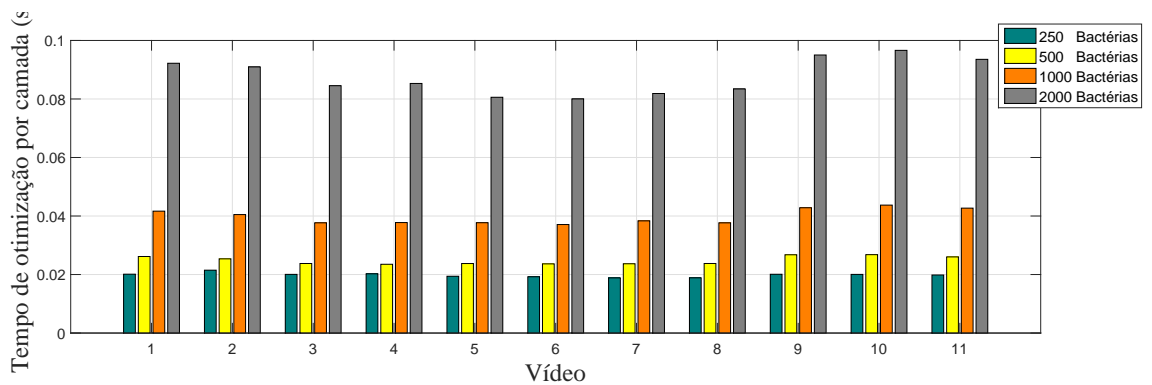


Figura 40: Tempo médio de otimização por camada de movimento obtido pelo FF.

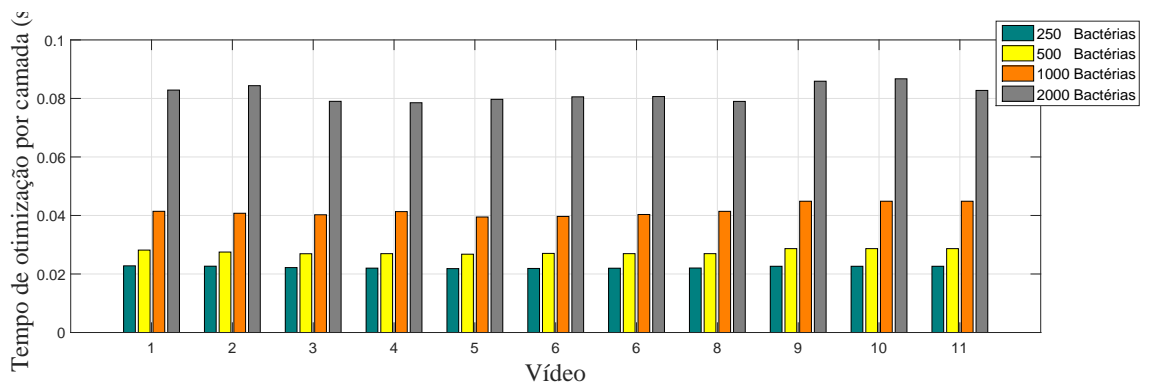


Figura 41: Tempo médio de otimização por camada de movimento obtido pela DAF.

A Figura 42 referente ao uso de 500 bactérias na população inicial, revela o histórico das colônias de bactérias geradas ao longo do vídeo 1. Este gráfico permite visualizar mais uma vez, a forma como as colônias de bactérias espelham o comportamento da multidão. Neste vídeo as pessoas caminham no ambiente nos primeiros instantes e repentinamente começam a correr em todas as direções. O momento do tumulto é caracterizado no gráfico como sendo aquele em que os valores de população, estoque e centroide se elevam repentinamente acima da média.

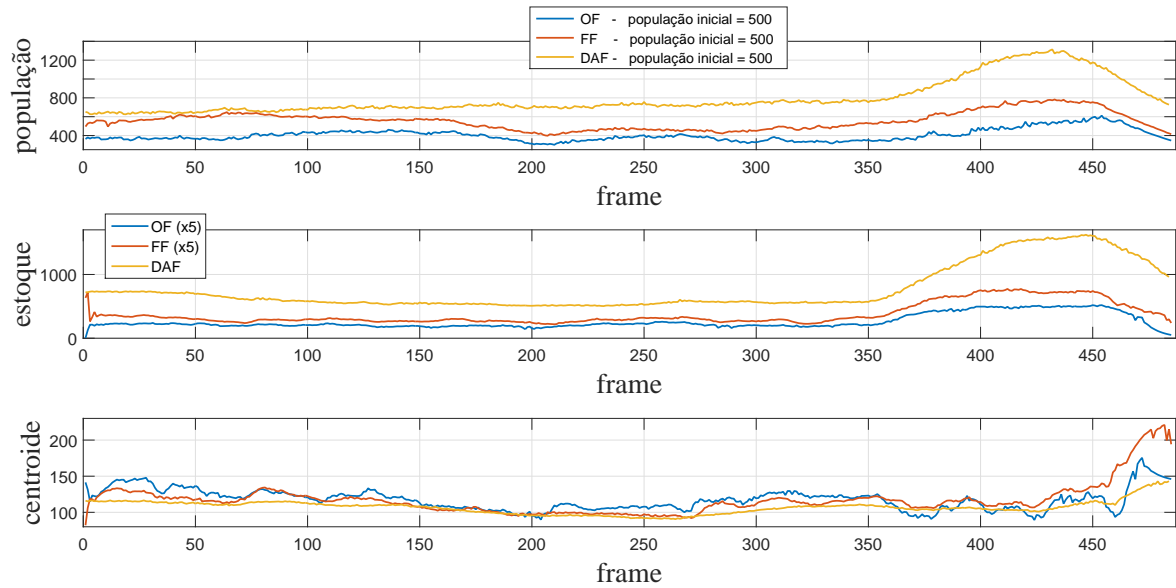


Figura 42: Registro das populações, estoques e centroides das colônias de bactérias obtidas ao longo da otimização do vídeo 1, para camadas de movimento geradas pelo FO, FF e DAF.

Para esse mesmo vídeo 1 percebem-se três eventos bem definidos visualmente: pessoas caminhando, em seguida as pessoas se dirigem para o centro do vídeo e repentinamente correm em fuga do centro para os cantos. Assim, definiu-se 3 neurônios para agrupar os dados. A Figura 43 mostra a posição final dos pesos após o treinamento dos mesmos dados apresentados na Figura 42, para o método de extração DAF. Para mostrar o espalhamento dos dados em 2 dimensões, foram utilizados os gráficos de população x estoque e centroide x estoque. É possível notar que há uma significativa distância entre o aglomerado de dados associados ao comportamento normal e os dados associados ao tumulto.

O número de neurônios definido para cada vídeo do *dataset* é apresentado na Tabela 6. Como na Tabela 3, mostra-se o tempo médio de treinamento para os vídeos, para a otimização com 250, 500, 1000 e 2000 bactérias iniciais, utilizando as 3 técnicas de extração de movimento. A última coluna da tabela mostra o tempo de processamento médio para a classificação de apenas uma amostra de dados após o treinamento da RNK.

A detecção dos eventos para o *dataset* UMN, mais uma vez em concordância com a observação humana e com trabalhos anteriores como (MEHRAN; OYAMA; SHAH, 2009; KAL TSA et al., 2014; RAO et al., 2013; LIU; LI; JIA, 2014; KAL TSA et al., 2015) é utilizada como referência para validar a classificação da RNK. A Figura 44 ilustra a comparação da detecção realizada pelo método proposto com a detecção esperada.

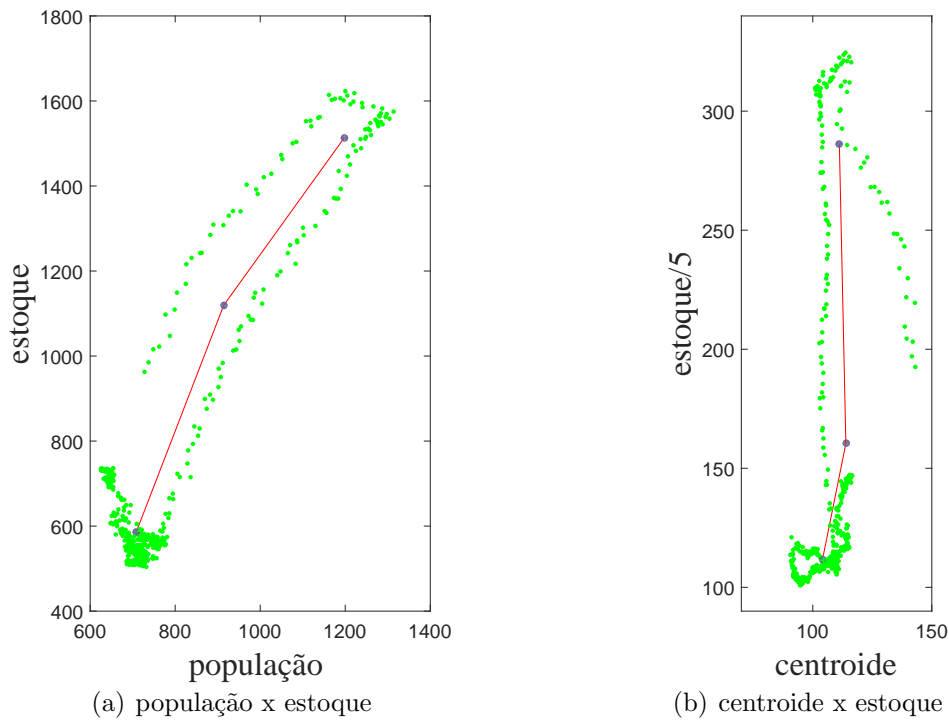


Figura 43: Distribuição dos neurônios da RNK para os registros de otimização do vídeo 1, para 500 bactérias iniciais e camadas de movimento extraídas pela DAF. Para melhor visualização da imagem, os valores de estoque foram divididos por 5 em (b).

Tabela 6: Número de neurônios, tempo médio de treinamento e desvio padrão, e tempo médio de classificação para uma amostra após o treino, para cada vídeo do *dataset* UMN

Vídeo	N ^o de neurônios	Treinamento (s)	Desvio padrão	Amostra (s)
1	3	2,7562	0,4120	1,1366e-06
2	4	2,8436	0,4678	2.2348e-06
3	5	2,3217	0,4678	1.4861e-06
4	4	3,5428	0,3628	2.1691e-06
5	3	2,1842	0,4578	1.9823e-06
6	3	2,3567	0,4231	1.7385e-06
7	3	2,1547	0,1937	9.1084e-07
8	3	2,7853	0,2825	2.3067e-06
9	4	3,0621	0,2445	2.0252e-06
10	3	2,2754	0,3584	2.3589e-06
11	4	3,2621	0,2665	2.4537e-06

Para avaliar a qualidade da classificação da RNK, a Figura 45 apresenta a curva ROC da detecção em todos os *frames* do *dataset* UMN, para os 3 métodos de extração de movimento. Nota-se que mais uma vez os valores encontram-se bem próximos, com uma pequena vantagem para o FF.

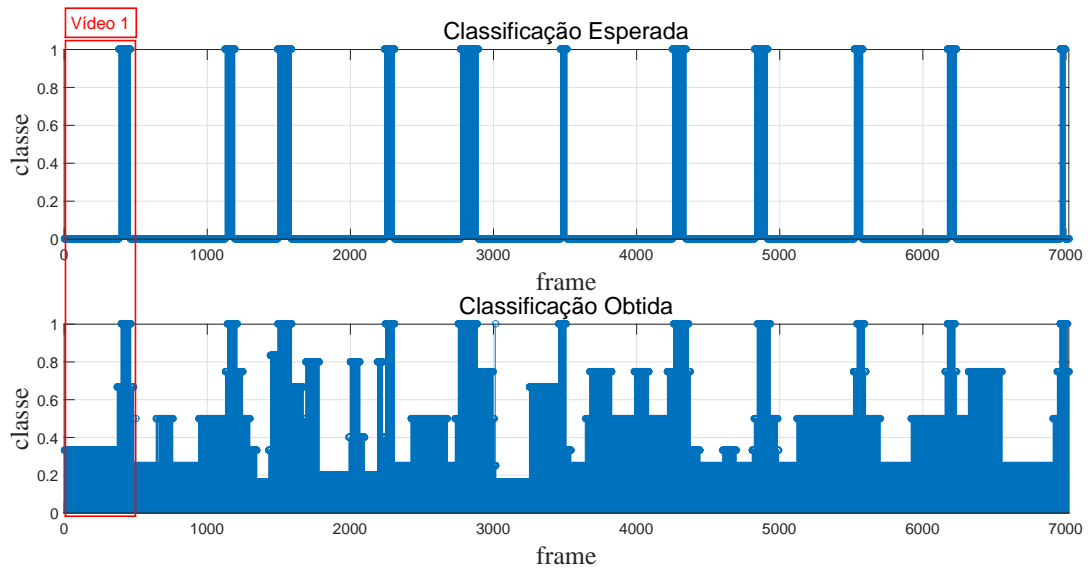


Figura 44: Comparação entre as classes desejadas, que possuem um evento considerado anormal, e as classes obtidas pelo método proposto para todos os frames do *dataset* UMN. Esta figura refere-se apenas ao resultado obtido utilizando a DAF para a extração de movimento e 500 bactérias iniciais na otimização.

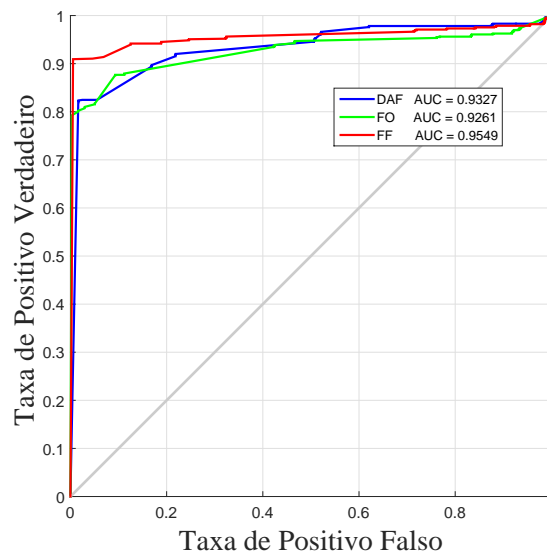


Figura 45: Curva ROC das detecções realizadas em todos os vídeos do *dataset* UMN utilizando as 3 diferentes técnicas de extração de movimento

A Tabela 7 expõe os resultados obtidos neste trabalho em contraste com a AUC obtida por outros métodos. O método proposto apresentou valores de AUC levemente inferiores aos dos sistemas referenciados, no entanto, exige muito menos esforço computacional.

Tabela 7: Área sob a curva ROC do método proposto *vs.* outros métodos para o *dataset* UMN

Método	Área sob a curva ROC	Melhoria
FO (MEHRAN; OYAMA; SHAH, 2009)	0,8400	–
MFS (MEHRAN; OYAMA; SHAH, 2009)	0,9600	14,28%
Invariante Caótica (WU; MOORE; SHAH, 2010)	0,9900	17,85%
Reconstrução Escassa (CONG; YUAN; LIU, 2011)	0,9780	16,42%
Bacteria Colony - OF	0,9262	10,26%
Bacteria Colony - FF (MSF)	0,9549	13,68%
Bacteria Colony - DAF	0,9327	11,04%

6.5 Video de Longa Duração

Os vídeos analisados nas seções anteriores possuem em média apenas alguns segundos de duração. O método proposto nesta dissertação busca, entretanto, uma sistematização que possa indicar caminhos para análises em vídeos de longa duração, ou em tempo real, para aplicações associadas a detecções de eventos catastróficos. Por isso, para avaliar o potencial do método, um vídeo de longa duração foi gravado especialmente para os experimentos. O vídeo possui duração de 61 minutos e foi particionado em 15 partes de aproximadamente 4 minutos cada. A taxa de *frames* foi reduzida para 4 *fps* devido à capacidade computacional disponível para realizar o processamento de imagens. Infelizmente não foi possível contemplar a avaliação com as técnicas de FO e FF, devido a exigência de elevado custo computacional na fase de pré-processamento. Deste modo, utiliza-se apenas a DAF para realizar a extração de movimento. As especificações do vídeo são mostradas na Tabela 8:

Tabela 8: Especificações do vídeo

Vídeo	Especificações
Duração	3660 <i>s</i>
Taxa de <i>frames</i>	4 <i>fps</i>
Número de <i>frames</i>	14.640
Largura	640
Altura	480
Tamanho por parte	8MB
Formato	.mp4

O vídeo registra o tráfego de carros no cruzamento da radial oeste em frente ao Estádio Jornalista Mário Filho, mais conhecido como Maracanã, no Rio de Janeiro. Este

local foi escolhido devido aos eventuais tumultos que ocorrem nos dias de jogos decisivos. Existem basicamente dois eventos distintos: semáforo direito aberto e semáforo direito fechado. A Figura 46 apresenta 2 *frames* evidenciando a distinção entre os eventos. Nota-se que quando o semáforo direito está fechado, o semáforo do outro lado da avenida encontra-se aberto, e vice versa.



(a) Semáforo da direita aberto



(b) Semáforo da direita fechado

Figura 46: *Frames* representando os dois eventos distintos considerados normais.

O vídeo possui um total de 14.640 *frames*, mas por questões de visualização do comportamento das bactérias na otimização, a Figura 47 apresenta um trecho contendo apenas os dados associados aos 5000 primeiros *frames*, para ilustrar em detalhes os resultados obtidos. Verifica-se que as populações e os centroides espelham satisfatoriamente os eventos. O estoque, no entanto, apresenta variações caóticas devido à presença de motocicletas, carros e ônibus em velocidades variadas na cena. Como o estoque é proporcional à quantidade de movimento, a imprevisibilidade dos veículos foi captada pelos padrões essencialmente caóticos formados pelos dados referentes ao estoque.

É interessante observar a repetição aproximada dos mesmos padrões apresentados pelas colônias de bactérias, que refletem a natureza cíclica dos eventos que ocorrem no vídeo. As pequenas variações nas curvas estão ligadas aos pequenos eventos no vídeo que não possuem significância comparados aos eventos dominantes.

A Figura 48 mostra o tempo médio de pré-processamento, em que ocorre a extração da camada de movimento utilizando apenas a DAF e o tempo médio da otimização associase as partes do vídeo que vão de P1 a P15.

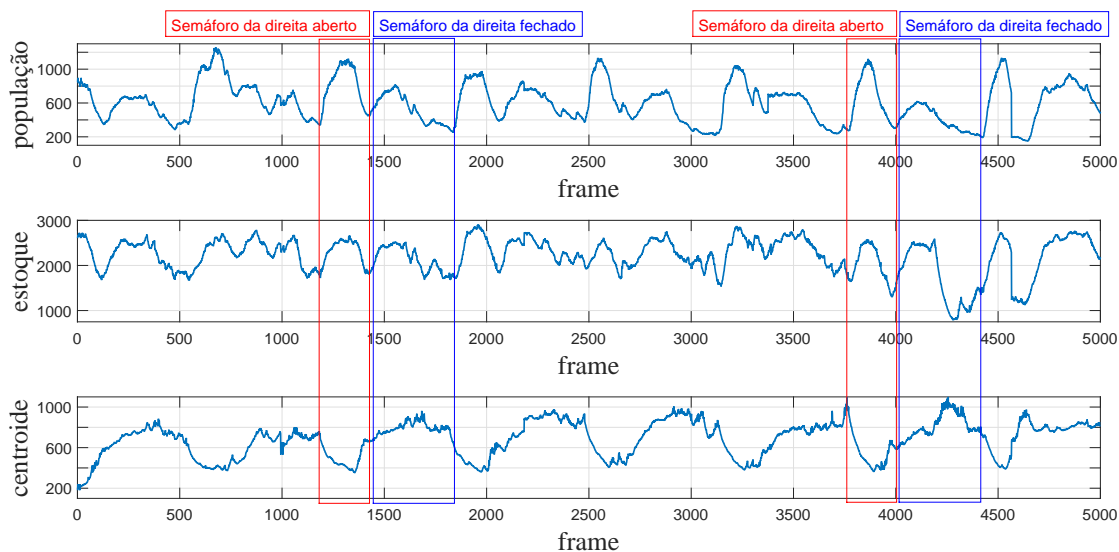


Figura 47: Comportamento das populações de bactérias, com destaque para os padrões que emergem da otimização.

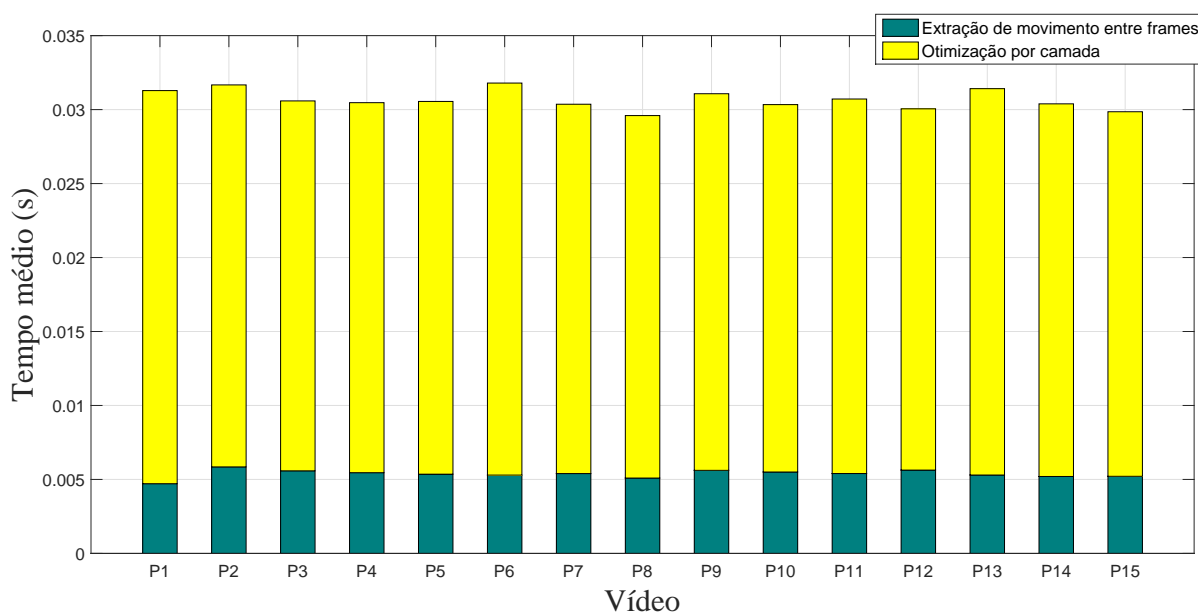


Figura 48: Tempo médio para a extração de movimento e otimização das camadas, para cada parte do vídeo.

A Figura 49 apresenta a distribuição bidimensional dos dados e a posição dos neurônios. Foram escolhidos 4 neurônios para compor a RNK: 2 para captar os eventos relacionados à abertura e fechamento do semáforo, 1 neurônio para indicar a transição entre esses dois eventos e mais 1 neurônio para captar os eventos imprevisíveis que podem ser caracterizados como anormalidades. A Figura 50 apresenta a distribuição tridimensional dos dados, revelando concentrações nas áreas correspondentes aos eventos interpretados como normais.

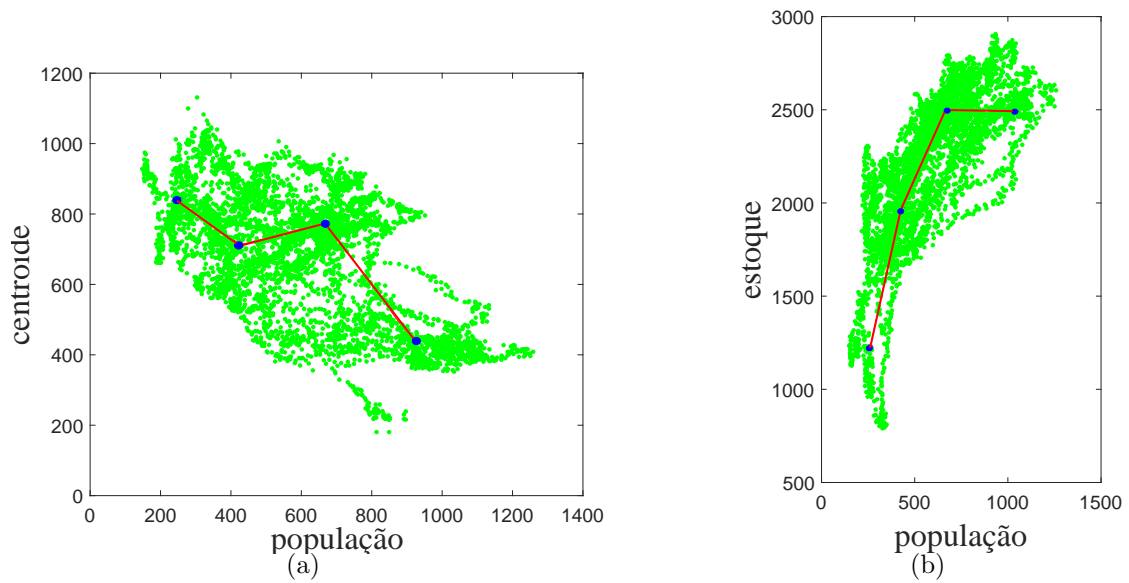


Figura 49: Distribuição dos neurônios da RNK para os registros de otimização de todas as partes do vídeo para 500 bactérias iniciais e camadas de movimento extraídas pela DAF. (a) revela a distribuição bidimensional da população x centroide e (b) a distribuição da população x estoque.

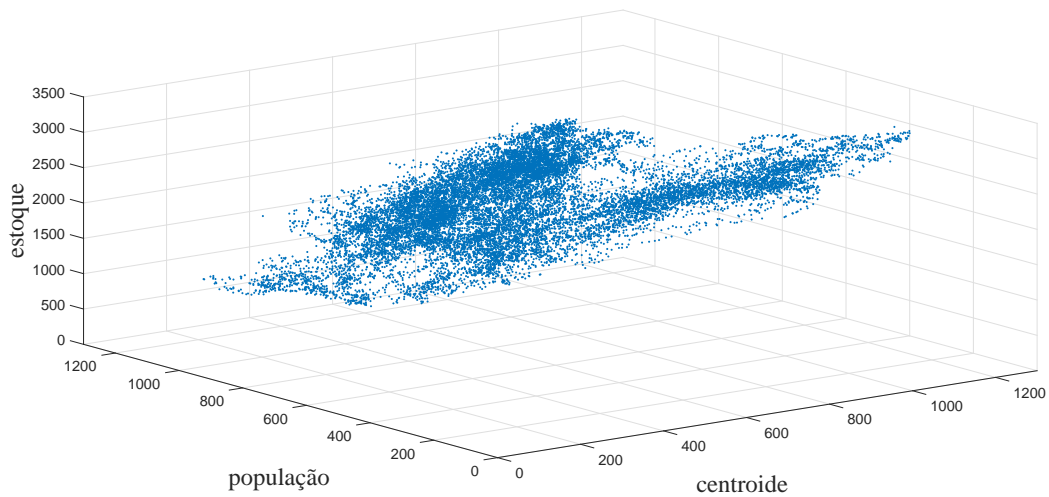


Figura 50: Visualização dos dados em 3 dimensões

A Figura 51 mostra como a RNK realizou a detecção após o treinamento, para os 5000 primeiros *frames* do vídeo. A classe 4 diz respeito ao semáforo direito aberto, a classe 3 ao semáforo direito fechado, a classe 2 representa a transição entre o evento da classe 4 e 3, finalmente a classe 1 está associada aos eventos inesperados que ocorreram na cena, que correspondem a dois pedestres atravessando fora da faixa e um pequeno congestionamento causado por veículos que obstruíram o cruzamento, como mostra a Figura 52.

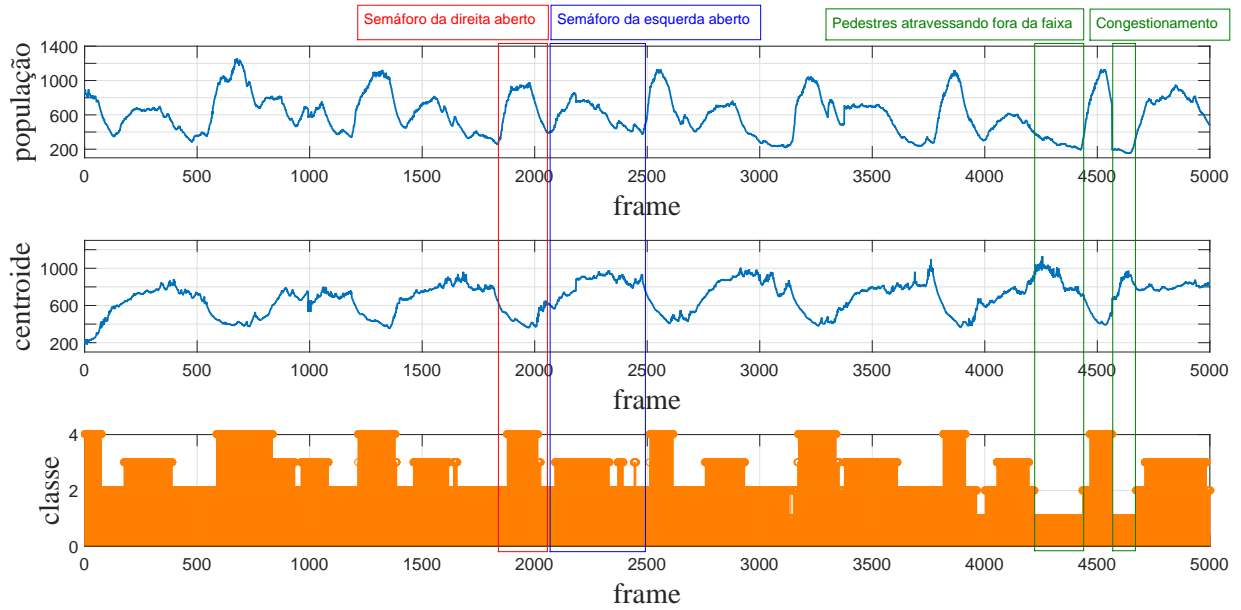


Figura 51: Detecção após o treinamento da rede.



(a) Pedestres atravessando fora da faixa

(b) Congestionamento

Figura 52: *Frames* correspondentes aos eventos destacados na Figura 51.

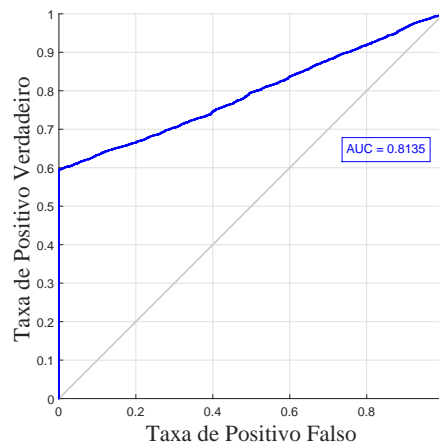


Figura 53: Curva ROC para a classificação de todos os *frames* do vídeo.

A Figura 53 apresenta a curva ROC de todos os *frames* do vídeo. Existe uma transição entre os dois eventos dominantes da cena, que a RNK identifica como uma nova classe. Devido à incerteza da classificação dos dados associados a essa região de transição, houve uma diminuição na taxa de detecções positivas verdadeiras, e assim, a área sob a curva ROC obtida foi de 0,8135, indicando que o sistema obteve uma boa detecção, mas necessita de melhorias na etapa de classificação dos dados.

6.6 Considerações Finais

Neste capítulo foi avaliada a capacidade do algoritmo CBA de otimizar as camadas de movimento obtidas pelas técnicas de DAF, FO e FF. Avaliou-se também o tempo de processamento por par de *frames* na etapa de geração das camadas de movimento, a qualidade das classificações realizadas pela RNK e a confiabilidade das detecções em relação aos métodos propostos em trabalhos anteriores. A robustez do sistema proposto foi verificada para um vídeo de longa duração, com o intuito de investigar a possibilidade de utilização do sistema em aplicações de vigilância em tempo real.

A técnica de extração de movimento baseada na DAF, utilizada em conjunto com o algoritmo CBA, mostrou resultados semelhantes aos encontrados pelas técnicas de FO e FF. A técnica de DAF, mesmo dotada de pouca sofisticação matemática, consegue atender satisfatoriamente o propósito de extrair o movimento dos *frames*, devido ao dinamismo e a adaptabilidade da otimização por CBA que praticamente desconsidera o ruído. A vantagem da utilização da DAF reside principalmente no tempo de processamento das imagens, que é consideravelmente menor do que os obtidos com o FO e o FF.

O algoritmo CBA nas configurações indicadas e para valores de população inicial de 250 e 500 apresentou resultados que revelam o seu potencial para a otimização de vídeos contendo cenários diversos. A leveza e simplicidade do algoritmo, aliada à sua capacidade adaptativa, permitiram que o tempo de otimização de praticamente todas as camadas de movimento dos experimentos, nas configurações supracitadas, permanecesse abaixo de 0,033, tempo este que correspondente a taxa de *frames* dos vídeos utilizados nos experimentos.

A classificação dos dados realizada pela RNK necessita de baixo custo computacional, no entanto, as taxas de detecção de verdadeiro positivo e falso positivo se mantiveram levemente abaixo daquelas obtidas por trabalhos anteriores.

Capítulo 7

CONCLUSÕES E TRABALHOS FUTUROS

NESTE capítulo são apresentadas as principais conclusões acerca do método de detecção de eventos em vídeos proposto nesta dissertação, bem como as conclusões relativas ao desempenho do algoritmo na otimização de camadas de movimento geradas pelas técnicas DAF, FO e FF. Adicionalmente, são apresentadas algumas possíveis melhorias em trabalhos futuros.

7.1 Conclusões

Esta dissertação abordou o problema de detecção de eventos em vídeos com multidões. O trabalho desenvolvido propõe um sistema que possibilita a detecção de tais eventos com baixo custo computacional e se mostrou capaz de realizar detecções em tempo real, com o uso da DAF e do algoritmo de CBA configurado para 250 e 500 bactérias na população inicial de cada camada. A motivação surgiu do intuito de tentar propor uma solução para o problema tecnológico recente de analisar grandes quantidades de imagens provenientes de câmeras de vigilância em tempo real, e assim prevenir ou atenuar os efeitos de possíveis eventos catastróficos.

O sistema proposto é dividido em 3 etapas básicas: pré-processamento, otimização e classificação. Realiza-se no pré-processamento a extração de movimento entre os *frames* na forma de camadas, em seguida ocorre a otimização para identificar as regiões de interesse de cada uma das camadas, e por último, tem-se a classificação dos registos comportamentais associados ao processo de otimização. O movimento entre *frames* é obtido pelas técnicas de diferença absoluta entre *frames* (DAF), fluxo óptico (FO) ou fluxo de forças (FF). A otimização é realizada pelo algoritmo de colônia de bactérias artificiais

(CBA). A classificação dos registros é feita pela rede neural de Kohonen (RNK), que por sua vez é treinada com os registros comportamentais das CBA durante a otimização.

Esta dissertação analisou os ganhos em eficiência computacional na fase de pré-processamento, comparando os tempos de extração de movimento pelas técnicas DAF, FO e FF. A técnica DAF apresentou os melhores resultados devido a sua extrema simplicidade matemática quando comparada as outras duas técnicas. No entanto, quando comparada ao FO e o FF, a DAF está mais propensa às influências de ruídos provenientes do vídeo e às repentinas mudanças de iluminação na cena. Os resultados mostraram que o tempo dispendido na extração de movimento pelo FO e FF varia de acordo com o tamanho dos volumes espaço-temporais processados, diferentemente da DAF, que apresentou maior estabilidade. Apesar da certa previsibilidade dos resultados, em que naturalmente uma técnica simples com a DAF supera a maioria das técnicas neste campo, os exaustivos e extensivos experimentos indicaram que o potencial da técnica foi bem explorado devido ao uso de CBA na otimização das camadas.

A otimização consiste em rapidamente alocar grandes quantidades de bactérias apenas nas regiões de interesse da camada, sinalizando a presença de movimento. Nessa otimização, o ruído possui baixa magnitude quando comparado àquelas associadas às regiões com movimento acima da média, e por isso, é desconsiderado pelo processo de seleção natural que fundamenta o algoritmo das CBA. Deste modo, as 3 técnicas de extração de movimento são passíveis de serem utilizadas com o algoritmo CBA, com valores médios de tempo de otimização muito próximos, já que a camada de movimento, independente de sua procedência, é para a CBA apenas uma função de custo.

As populações de bactérias, o estoque de aptidão acumulado por estas ao longo das épocas evolutivas de uma camada e o centroide das colônias revelam a natureza dos acontecimentos da cena. Os padrões de comportamento apresentados pelas bactérias na otimização, quando submetidas às cenas contendo os mesmos eventos, são identificados e classificados rapidamente pela RNK. Essas redes, cujo algoritmo de treinamento é também baseado em um método competitivo, realizam o agrupamento dos dados e permitem uma classificação com margens que se harmonizam com a imprecisão do comportamento humano em vídeo. Entretanto, tal imprecisão não impacta na confiabilidade das detecções do sistema proposto. A qualidade das detecções é mensurada pela área sob a curva ROC, e os valores encontrados nesta dissertação estão próximos dos resultados obtidos pelas

técnicas referenciadas nas Tabelas 4 e 7, relacionados aos *datasets* UMN e PETS 2009, respectivamente.

Constatou-se nos experimentos utilizando o vídeo de longa duração, que o sistema proposto foi capaz realizar a detecção de eventos anormais, tomando como referência os comportamentos cíclicos esperados na cena. Para a extração de movimento pela DAF, e com uma população inicial de bactérias de 250 e 500 na primeira época de cada camada, o tempo de processamento do sistema de detecção implementado esteve sempre abaixo de 0,033 segundos por par de *frames*. Existem dois eventos dominantes no vídeo: semáforo direito aberto ou fechado. No entanto, existe uma transição entre os dois eventos, que a rede neural de Kohonen identifica como uma nova classe. Devido à incerteza da classificação dos dados associados a essa região de transição, houve uma diminuição na taxa de detecções positivas verdadeiras, e assim, a área sob a curva ROC obtida foi de 0,8135. Quanto aos eventos anormais presentes nas cenas, o sistema foi capaz de realizar a detecção com precisão.

7.2 Trabalhos Futuros

Nesta Seção, sugerem-se algumas formas de continuar o estudo do tema abordado. Com relação à extração de movimento entre *frames*, outras técnicas podem ser utilizadas na tentativa de obter camadas que retratem com maior fidelidade, a quantidade e a localização dos movimentos. Seria vantajoso se tais experimentos tivessem como objetivo tanto a melhoria da captação de movimento, quando a redução do custo computacional.

Melhorias também podem ser implementadas no algoritmo das CBA. A natureza, através da competição e seleção natural, é o otimizador perfeito, por isso a aproximação do algoritmo com comportamento real das bactérias, torna a otimização cada vez mais leve e precisa. A utilização de uma funcionalidade, representando o DNA, poderia identificar cada bactéria e suas descendentes, fornecendo assim informações extras para facilitar o agrupamento dos dados. A implementação deste DNA deve ser feita de tal modo que não impacte significativamente no tempo de otimização.

O sistema de classificação, apesar de apresentar resultados satisfatórios, precisa ser melhorado em alguns aspectos para que haja um incremento de eficiência. Nesse sentido, um aumento do número de neurônios representa um acréscimo de precisão que não elevaria a um patamar proibitivo o tempo global da detecção. No entanto, quando

maior o número de neurônios, maior será o esforço humano para associar as classes ao número de eventos.

Para a utilização do sistema proposto em aplicações de vigilância visando realizar a detecção ou a predição de anormalidades em um cenário específico, é necessário que a rede neural de Kohonen seja treinada com dados provenientes de vídeos de longa duração. Quanto maior a duração dos vídeos, maior será a precisão das classificações da rede.

REFERÊNCIAS

AGARWAL, A.; GUPTA, S.; SINGH, D. K. Review of optical flow technique for moving object detection. In: IEEE. *Contemporary Computing and Informatics (IC3I), 2016 2nd International Conference on*. Delhi, India, 2016. p. 409–413.

ANDERSON, K.; MCOWAN, P. W. A real-time automated system for the recognition of human facial expressions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 36, n. 1, p. 96–105, 2006.

AUGEREAU, B.; TREMBLAIS, B.; FERNANDEZ-MALOIGNE, C. Vectorial computation of the optical flow in color image sequences. In: SOCIETY FOR IMAGING SCIENCE AND TECHNOLOGY. *Color and Imaging Conference*. Arizona, USA, 2005. v. 2005, n. 1, p. 130–134.

BARRON, J. L.; FLEET, D. J.; BEAUCHEMIN, S. S. Performance of optical flow techniques. *International journal of computer vision*, Springer, v. 12, n. 1, p. 43–77, 1994.

BILAL, M.; KHAN, S. Xilinx fpgas advance autonomous monitoring of crowds. *Xcelljournal*, Xilinx, v. 93, p. 14–21, 2015.

BLACK, M. J.; ANANDAN, P. A framework for the robust estimation of optical flow. In: IEEE. *Computer Vision, 1993. Proceedings., Fourth International Conference on*. Berlin, Germany, 1993. p. 231–236.

BONABEAU, E.; DORIGO, M.; THERAULAZ, G. *Swarm intelligence: from natural to artificial systems*. Oxford, Reino Unido: Oxford university press, 1999.

BREMERMANN, H. Chemotaxis and optimization. *Journal of the Franklin Institute*, Elsevier, v. 297, n. 5, p. 397–404, 1974.

- CANDAMO, J. et al. Understanding transit scenes: A survey on human behavior-recognition algorithms. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 11, n. 1, p. 206–224, 2010.
- CAO, T. et al. Abnormal crowd motion analysis. In: IEEE. *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*. Pequim, China, 2009. p. 1709–1714.
- CHAARAOU, A. A.; CLIMENT-PÉREZ, P.; FLÓREZ-REVUELTA, F. A review on vision techniques applied to human behaviour analysis for ambient-assisted living. *Expert Systems with Applications*, Elsevier, v. 39, n. 12, p. 10873–10888, 2012.
- CHEN, D.-Y.; HUANG, P.-C. Motion-based unusual event detection in human crowds. *Journal of Visual Communication and Image Representation*, Elsevier, v. 22, n. 2, p. 178–186, 2011.
- CIVICIOGLU, P. Artificial cooperative search algorithm for numerical optimization problems. *Information Sciences*, Elsevier, v. 229, p. 58–76, 2013.
- CONG, Y.; YUAN, J.; LIU, J. Sparse reconstruction cost for abnormal event detection. In: IEEE. *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. Colorado Springs, CO, USA, 2011. p. 3449–3456.
- COSTA, J. R. d.; NEDJAH, N.; MOURELLE, L. M. d. Crowd anomaly detection based on optical flow, artificial bacteria colony and kohonen neuralnetwork. In: IEEE. *Computational Science and Its Applications, ICCSA 2017*. Trieste, Itália, 2017. p. 1–16.
- COSTA, J. R. d.; NEDJAH, N.; MOURELLE, L. M. d. *Crowd Anomaly Detection Using Artificial Bacteria Colony*. 2017. aceito para publicação em Springer, Multimedia Tools and Applications .
- DORIGO, M.; STÜTZLE, T. Ant colony optimization: overview and recent advances. In: *Handbook of metaheuristics*. : Springer, 2010. p. 227–263.
- EIBL, G.; BRANDLE, N. Evaluation of clustering methods for finding dominant optical flow fields in crowded scenes. In: IEEE. *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. Tampa, USA, 2008. p. 1–4.

- EISENBACH, M. Functions of the flagellar modes of rotation in bacterial motility and chemotaxis. *Molecular microbiology*, Wiley Online Library, v. 4, n. 2, p. 161–167, 1990.
- ENGELBRECHT, A. P. *Computational intelligence: an introduction*. : John Wiley & Sons, 2007.
- FANG, Y.; LIU, Y.; LIU, J. A novel simplified foraging optimization algorithm for parameter identification of nonlinear system model. In: IEEE. *Automation and Logistics, 2007 IEEE International Conference on*. Jinan, China, 2007. p. 798–802.
- FENG, J.; ZHANG, C.; HAO, P. Online learning with self-organizing maps for anomaly detection in crowd scenes. In: IEEE. *Pattern Recognition (ICPR), 2010 20th International Conference on*. Istanbul, Turkey, 2010. p. 3599–3602.
- FERREIRA, M. Detecção de movimento através de subtração de fundo para vigilância eletrônica automática. *Proposta de Tese (Doutorado) Doutorado em Informática, PUC-Rio, Rio de Janeiro, RJ*, 2007.
- GAO, Y. et al. A modified social force model considering relative velocity of pedestrians. In: IEEE. *2013 IEEE International Conference on Automation Science and Engineering (CASE)*. Madison, WI, USA, 2013. p. 747–751.
- GAZI, V.; PASSINO, K. M. Stability analysis of swarms in an environment with an attractant/repellent profile. In: IEEE. *American Control Conference, 2002. Proceedings of the 2002*. Alaska, USA, 2002. v. 3, p. 1819–1824.
- GROETSCH, C. The theory of tikhonov regularization for fredholm equations. *104p, Boston Pitman Publication*, 1984.
- GUNEY, K.; BASBUG, S. Interference suppression of linear antenna arrays by amplitude-only control using a bacterial foraging algorithm. *Progress In Electromagnetics Research*, EMW Publishing, v. 79, p. 475–497, 2008.
- HASSAN, M. A. et al. Foreground extraction for real-time crowd analytics in surveillance system. In: IEEE. *Consumer Electronics (ISCE 2014), The 18th IEEE International Symposium on*. JeJu Island, Korea (South), 2014. p. 1–2.

- HELBING, D.; MOLNAR, P. Social force model for pedestrian dynamics. *Physical review E*, APS, v. 51, n. 5, p. 4282, 1995.
- HELBING, D. et al. Self-organizing pedestrian movement. *Environment and planning B: planning and design*, SAGE Publications Sage UK: London, England, v. 28, n. 3, p. 361–383, 2001.
- HORN, B. K.; SCHUNCK, B. G. Determining optical flow. *Artificial intelligence*, Elsevier, v. 17, n. 1-3, p. 185–203, 1981.
- HUANG, J.-F.; CHEN, S.-L. Detection of violent crowd behavior based on statistical characteristics of the optical flow. In: IEEE. *Fuzzy Systems and Knowledge Discovery (FSKD), 2014 11th International Conference on*. Xiamen, China, 2014. p. 565–569.
- JADHAV, D. G.; DEVI, S.; PATTNAIK, S. S. Memetic algorithm with pso or bfo's chemotaxis mechanism as local search for function optimization. In: IEEE. *Recent Advances in Computing and Software Systems (RACSS), 2012 International Conference on*. Chennai, India, 2012. p. 42–47.
- KAJO, I.; MALIK, A. S.; KAMEL, N. Motion estimation of crowd flow using optical flow techniques: A review. In: IEEE. *Signal Processing and Communication Systems (ICSPCS), 2015 9th International Conference on*. Cairns, Australia, 2015. p. 1–9.
- KAL TSA, V. et al. Swarm-based motion features for anomaly detection in crowds. In: IEEE. *2014 IEEE International Conference on Image Processing (ICIP)*. Paris, France, 2014. p. 2353–2357.
- KAL TSA, V. et al. Swarm intelligence for detecting interesting events in crowded environments. *IEEE transactions on image processing*, IEEE, v. 24, n. 7, p. 2153–2166, 2015.
- KARABOGA, D. et al. A comprehensive survey: artificial bee colony (abc) algorithm and applications. *Artificial Intelligence Review*, Springer, v. 42, n. 1, p. 21–57, 2014.
- KARLSSON, J. B. S. 2016. tutorial and toolbox on real time optical flow. Disponível em: <<http://www.mathworks.com/matlabcentral/fileexchange/44400-tutorial-and-toolbox-on-real-time-optical-flow>>.

- KHILAR, R.; CHITRAKALA, S. A novel technique for efficient extraction of foreground and analysis of road traffic flow. In: IEEE. *Science Engineering and Management Research (ICSEMR), 2014 International Conference on*. Chennai, India, 2014. p. 1–5.
- KIM, D. H.; CHO, J. H. A biologically inspired intelligent pid controller tuning for avr systems. *International Journal of Control Automation and Systems*, Korean Institute of Electrical Engineers, Wuhan, China, v. 4, n. 5, p. 624, 2006.
- KIM, G.-J. et al. Automated measurement of crowd density based on edge detection and optical flow. In: IEEE. *Industrial Mechatronics and Automation (ICIMA), 2010 2nd International Conference on*. Wuhan, China, 2010. v. 2, p. 553–556.
- KIRCHNER, A.; SCHADSCHNEIDER, A. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A: statistical mechanics and its applications*, Elsevier, v. 312, n. 1, p. 260–276, 2002.
- KOHONEN, T. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, Springer, v. 43, n. 1, p. 59–69, 1982.
- KOHONEN, T. *Self-organization and associative memory*. : Springer Science & Business Media, 2012.
- LI, M. et al. Bacterial foraging algorithm with varying population for optimal power flow. In: SPRINGER. *Workshops on Applications of Evolutionary Computation*. London, England, 2007. p. 32–41.
- LIN, S.; LIU, Y.; FANG, Y. Weighted fuzzy rule-based system combined with a novel simplified e. coli foraging optimization algorithm. In: IEEE. *Intelligent Control, 2007. ISIC 2007. IEEE 22nd International Symposium on*. Suntec Convention Centre, Singapore, 2007. p. 267–270.
- LIU, Y.; LI, X.; JIA, L. Abnormal crowd behavior detection based on optical flow and dynamic threshold. In: IEEE. *Intelligent Control and Automation (WCICA), 2014 11th World Congress on*. Shenyang, China, 2014. p. 2902–2906.
- LIU, Y.; PASSINO, K. Biomimicry of social foraging bacteria for distributed optimization: models, principles, and emergent behaviors. *Journal of optimization theory and applications*, Springer, v. 115, n. 3, p. 603–628, 2002.

- LUCAS, B. D.; KANADE, T. et al. An iterative image registration technique with an application to stereo vision. In: *IJCAI*. Vancouver, Canada: Intl Joint Conference on Artificial Intelligence, 1981. v. 81, n. 1, p. 674–679.
- MARANA, A. et al. On the efficacy of texture analysis for crowd monitoring. In: *IEEE. Computer Graphics, Image Processing, and Vision, 1998. Proceedings. SIBGRAP'98. International Symposium on*. Rio de Janeiro, Brazil, 1998. p. 354–361.
- MARTONE, R. et al. Comparative analysis between modern heuristics and hybrid algorithms. *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, Emerald Group Publishing Limited, v. 26, n. 2, p. 259–268, 2007.
- MEHRAN, R.; OYAMA, A.; SHAH, M. Abnormal crowd behavior detection using social force model. In: *IEEE. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Miami, USA, 2009. p. 935–942.
- MISHRA, S. A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 9, n. 1, p. 61–73, 2005.
- MONTAGUE, G.; WARD, A. A sub-optimal solution to the optimisation of bioreactors using the chemotaxis algorithm. *Process Biochemistry*, Elsevier, v. 29, n. 6, p. 489–496, 1994.
- MULLER, S. D. et al. Optimization based on bacterial chemotaxis. *IEEE transactions on Evolutionary Computation*, IEEE, v. 6, n. 1, p. 16–29, 2002.
- MURRAY, J. D. *Mathematical biology I: an introduction, Vol. 17 of interdisciplinary applied mathematics*. New York, USA: Springer, 2002.
- NGUYEN, H. T.; BHANU, B. Real-time pedestrian tracking with bacterial foraging optimization. In: *IEEE. Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*. Beijing, China, 2012. p. 37–42.
- NURMAINI, S. et al. Pattern recognition approach for swarm robots reactive control with fuzzy-kohonen networks and particle swarm optimization algorithm. *Journal*

of *Telecommunication, Electronic and Computer Engineering (JTEC)*, v. 8, n. 3, p. 155–160, 2016.

PANIGRAHI, B.; PANDI, V. R. Bacterial foraging optimisation: Nelder–mead hybrid algorithm for economic load dispatch. *IET Generation, Transmission & Distribution*, IET, v. 2, n. 4, p. 556–565, 2008.

PAPENBERG, N. et al. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, Springer, v. 67, n. 2, p. 141–158, 2006.

PASSINO, K. M. *Biomimicry for optimization, control, and automation*. : Springer Science & Business Media, 2005.

PASSINO, K. M. Bacterial foraging optimization. *Innovations and Developments of Swarm Intelligence Applications*, IGI Global, p. 219–233, 2012.

PETS, . Pets 2009 benchmark data disponível em <http://www.cvg.rdg.ac.uk/pets2009/a.html>. In: . Miame, Florida: [s.n.], 2009.

PLA, F. et al. Extracting motion features for visual human activity representation. *Pattern Recognition and Image Analysis*, Springer, p. 229–252, 2005.

RAGHAVENDRA, R. et al. Optimizing interaction force for global anomaly detection in crowded scenes. In: IEEE. *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. Barcelona, Spain, 2011. p. 136–143.

RAO, A. S. et al. Crowd density estimation based on optical flow and hierarchical clustering. In: IEEE. *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*. Mysore, India, 2013. p. 494–499.

RAO, A. S. et al. Crowd event detection on optical flow manifolds. *IEEE Transactions on Systems*, Hong Kong, China, 2015.

RASHWAN, H. A.; PUIG, D.; GARCIA, M. A. Improving the robustness of variational optical flow through tensor voting. *Computer Vision and Image Understanding*, Elsevier, v. 116, n. 9, p. 953–966, 2012.

- RITTER, H. et al. *Neural computation and self-organizing maps: an introduction.* : Addison-Wesley Reading, MA, 1992.
- SAADI, S. et al. Bacterial foraging algorithm for neutron radiography image quality improvement. In: IEEE. *Systems Signals and Devices (SSD), 2010 7th International Multi-Conference on.* Amman, Jordan, 2010. p. 1–6.
- SEESE, N. et al. Adaptive foreground extraction for deep fish classification. In: IEEE. *Computer Vision for Analysis of Underwater Imagery (CVAUI), 2016 ICPR 2nd Workshop on.* Cancun, Mexico, 2016. p. 19–24.
- SHI, Y.; EBERHART, R. A modified particle swarm optimizer. In: IEEE. *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on.* Alaska, USA, 1998. p. 69–73.
- SILVA, I. d.; SPATTI, D. H.; FLAUZINO, R. A. Redes neurais artificiais para engenharia e ciências aplicadas. *São Paulo: Artliber*, p. 33–111, 2010.
- SPIES, H.; SCHARR, H. Accurate optical flow in noisy image sequences. In: IEEE. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on.* Vancouver, Canada, 2001. v. 1, p. 587–592.
- STAUFFER, C.; GRIMSON, W. E. L. Adaptive background mixture models for real-time tracking. In: IEEE. *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* Fort Collins, USA, 1999. v. 2, p. 246–252.
- STEINHAUS, H. Sur la division des corp materiels en parties. *Bull. Acad. Polon. Sci*, v. 1, n. 804, p. 801, 1956.
- THIDA, M. et al. Learning video manifolds for content analysis of crowded scenes. *Information and Media Technologies*, Information and Media Technologies, v. 7, n. 3, p. 1117–1123, 2012.
- THIDA, M.; ENG, H.-L.; REMAGNINO, P. Laplacian eigenmap with temporal constraints for local abnormality detection in crowded scenes. *IEEE Transactions on Cybernetics*, IEEE, v. 43, n. 6, p. 2147–2156, 2013.

- TORTORA, G. J. et al. *Microbiology: an introduction*. San Francisco, USA: Benjamin Cummings, 2004.
- TZIAKOS, I.; CAVALLARO, A.; XU, L.-Q. Event monitoring via local motion abnormality detection in non-linear subspace. *Neurocomputing*, Elsevier, v. 73, n. 10, p. 1881–1891, 2010.
- UMN. Unusual crowd activity dataset of university of minnesota, disponível em <http://mha.cs.umn.edu/movies/crowdactivity-all.avi>. In: . Minnessota: 2006, 2006.
- VESANTO, J.; ALHONIEMI, E. Clustering of the self-organizing map. *IEEE Transactions on neural networks*, Ieee, v. 11, n. 3, p. 586–600, 2000.
- WANG, Y.; WANG, D. Foreground extraction based on anomaly detection. *Electronics Letters*, IET, v. 50, n. 8, p. 593–595, 2014.
- WU, S.; MOORE, B. E.; SHAH, M. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In: IEEE. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. Sam Francisco, USA, 2010. p. 2054–2060.
- WU, S.; WONG, H. S. Crowd motion partitioning in a scattered motion field. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 42, n. 5, p. 1443–1454, 2012.
- WU, Z.; ERDAN, G.; CHUN, C. Extraction foreground in moving image. In: IEEE. *Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on*. Pula, Croatia, 2001. p. 272–276.
- WU, Z.; ZHENG, H.; WANG, J. Pedestrian counting based on crowd density estimation and lucas-kanade optical flow. In: IEEE. *Image and Graphics (ICIG), 2013 Seventh International Conference on*. Qingdao, China, 2013. p. 471–476.
- XIONG, G. et al. An energy model approach to people counting for abnormal crowd behavior detection. *Neurocomputing*, Elsevier, v. 83, p. 121–135, 2012.
- XU, M. et al. misfm: On combination of mutual information and social force model towards simulating crowd evacuation. *Neurocomputing*, Elsevier, v. 168, p. 529–537, 2015.

- YAN, G.-w.; LI, C.-q.; MU-CHAO, L. A novel swarm optimization algorithm based on social force model for multimodal functions. In: IEEE. *Control and Decision Conference (CCDC), 2012 24th Chinese*. Taiyuan, China, 2012. p. 849–854.
- YOUNG, K. D. The selective value of bacterial shape. *Microbiology and molecular biology reviews*, Am Soc Microbiol, v. 70, n. 3, p. 660–703, 2006.
- ZHAN, B. et al. Crowd analysis: a survey. *Machine Vision and Applications*, Springer, v. 19, n. 5, p. 345–357, 2008.
- ZHAO, J. et al. Crowd instability analysis using velocity-field based social force model. In: IEEE. *Visual Communications and Image Processing (VCIP), 2011 IEEE*. Tainan, Taiwan, 2011. p. 1–4.
- ZITOUNI, M. S. et al. Advances and trends in visual crowd analysis: A systematic survey and evaluation of crowd modelling techniques. *Neurocomputing*, Elsevier, v. 186, p. 139–159, 2016.