



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Rastreamento de Indivíduos em Sistema de Monitoramento

Ivo Sócrates Moraes de Oliveira

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Informática

Orientador

Prof. Dr. Bruno Luigi Macchiavello Espinoza

Coorientador

Prof. Dr. Flávio de Barros Vidal

Brasília

2013

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Programa de pós-graduação em Informática

Coordenador: Prof. Dr. Ricardo Pezzuol Jacobi

Banca examinadora composta por:

Prof. Dr. Bruno Luiggi Macchiavello Espinoza (Orientador) — CIC/UnB

Prof.^a Dr.^a Mylene Christine Queiroz de Farias — CIC/UnB

Prof.^a Dr.^a Carla Silva Rocha Aguiar — FGA/UnB

CIP — Catalogação Internacional na Publicação

Oliveira, Ivo Sócrates Moraes de.

Rastreamento de Indivíduos em Sistema de Monitoramento / Ivo Sócrates Moraes de Oliveira. Brasília : UnB, 2013.

90 p. : il. ; 29,5 cm.

Dissertação (Mestrado) — Universidade de Brasília, Brasília, 2013.

1. rastreamento, 2. otimização por enxame de partículas, 3. função linear discriminante, 4. histograma rgb.

CDU 004

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Rastreamento de Indivíduos em Sistema de Monitoramento

Ivo Sócrates Moraes de Oliveira

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Informática

Prof. Dr. Bruno Luigi Macchiavello Espinoza (Orientador)

CIC/UnB

Prof.ª Dr.ª Mylene Christine Queiroz de Farias

CIC/UnB

Prof.ª Dr.ª Carla Silva Rocha Aguiar

FGA/UnB

Prof. Dr. Ricardo Pezzuol Jacobi

Coordenador do Programa de pós-graduação em Informática

Brasília, 25 de julho de 2013

Dedicatória

Dedico este trabalho à Deus, autor e consumidor da minha fé!

Agradecimentos

Ao longo do processo de construção do conhecimento aplicado nesta obras, muitos amigos, colegas e familiares contribuíram direta ou indiretamente para o mesmo. Entre eles destacam-se minha amada esposa, Letícia Oliveira, pelo amor e cumplicidade, meus pais, Ary Oliveira e Ana Tereza, pela força e palavras de incentivo, meus queridos irmãos, Ary Henrique e Bruno Tácito, pelo apoio incondicional e os conselhos, meu sogros e sogra, Luiz e Alice Pinto, que com muito carinho cuidou do que possuo de mais precioso, minha querida tia Etelvina Oliveira e meus primos e primas, Elaene Leila, Goiran, Margareth e Wadja, que me receberam como muito carinho em seus lares e por, inúmeras vezes, me levar e buscar no Aeroporto e Rodoviária de Brasília, aos meus concunhados, Antônio Carlos e Rodrigo, e cunhadas, Adriana, Leidiane e Caroline, pelo apoio incondicional.

Agradeço especialmente ao meu orientador e mentor professor Bruno Macchiavello, que investiu seus conhecimentos e seu tempo, apontou os melhores caminhos e pacientemente me orientou. E também, agradeço ao meu co-orientador professor Flávio Vidal pelo apoio e conhecimentos repassados.

Agradeço aos amigos Antônio Marcos e Wosley Arruda, pela força e apoio em todos os momentos.

Agradeço também aos demais colegas das disciplinas do mestrado e aos professores(as): Alba Melo, Alexandre Zaghetto, Camilo Dorea, Flávio Moura, Maria Emília e Maurício Ayala pelo repasse de conhecimentos, competência e pelas sugestões e críticas.

Às professoras que participaram da banca, Mylene Farias e Carla Aguiar, pelas valiosas contribuições no resultado final desta obra.

Aos colegas de trabalho pela compreensão e apoio.

E não posso deixar de agradecer ao meu filho amado, que foi enviado por Deus para alegrar os meus dias e veio para completar a minha família. Agradeço a ele por sempre selecionar o seu sorriso mais lindo e agradável para me receber, mesmo após dias distantes.

Resumo

A monitoração eletrônica baseada em vídeo digital tem se tornado chave para a eficácia de diversas atividades, pois permite a identificação eficiente de anomalias no local monitorado, a identificação de indivíduos suspeitos e o esclarecimento de fatos, entre outras atividades. Portanto, surge a necessidade de algoritmos de rastreamento visual de baixa complexidade, que visa oferecer a capacidade de identificação da trajetória de indivíduos em sistemas de monitoração eletrônica em ambientes restritos. Esses algoritmos podem permitir uma compactação diferenciada em uma região de interesse, permitindo melhor desempenho na maioria dos padrões de codificação de vídeo, como o *High Efficiency Video Coding* (HEVC). Este trabalho propõe um algoritmo de rastreamento que utiliza o método de Otimização por Enxame de Partículas (PSO - *Particle Swarm Optimization*) com uma função de custo calculada por uma Função Discriminante Linear (LDF - *Linear Discriminant Function*), que utiliza histograma RGB (*Red, Green and Blue*) dos blocos de cada partícula para caracterização do objeto alvo. De forma sucinta, o algoritmo desenvolvido realiza, após o segundo quadro capturado, a detecção do objeto alvo, através da estimação de movimento e obtenção do fluxo óptico. Este processo é realizado para o treinamento da função de custo do método de PSO. Como mencionado anteriormente, a função de custo foi desenvolvida utilizando Funções Discriminantes Lineares, que são treinadas com base em histogramas RGB de blocos sobre o quadro atual de cada partícula para a caracterização do objeto alvo. A partir das características são formadas duas classes a alvo e a não alvo. Logo em seguida, é chamado um novo quadro que terá o alvo rastreado através do método de PSO, que se baseia em três elementos essenciais, a inércia, melhor posição local e melhor posição global. Estes elementos são utilizados para atualizar o deslocamento do enxame e, consequentemente, acompanhar o alvo. As atualizações da melhor posição local e melhor posição global são definidas pela avaliação da proximidade obtida entre o valor atual e o centroide da classe alvo obtido durante o treinamento. Através dos testes do foram identificadas as seguintes características do algoritmo proposto: rápida convergência, pois foram obtidos bons resultados no algoritmo com poucas iterações no método de PSO; baixo custo computacional, se comparado com métodos determinísticos comuns, pois realiza uma quantidade menor de operações; capacidade de tratar oclusões que não superam um quarto ($1/4$) da resolução do vídeo e capacidade satisfatória de rastreamento de objeto com movimentos arbitrários e abruptos.

Palavras-chave: rastreamento, otimização por enxame de partículas, função linear discriminante, histograma rgb.

Abstract

Electronic monitoring-based on digital video has become a key element to the effectiveness of several activities, such as an efficient identification of anomalies in the monitored environment, the identification of suspects and clarification of facts, among others. Therefore, the need of visual tracking algorithms with low computational complexity that allow identification of the subjects' trajectory in electronic monitoring systems has increase. These algorithms can allow differential coding in different image regions, allowing better performance in most video coding standard, as the High Efficiency Video Coding (HEVC). In this work it is propose an algorithm that uses Particle Swarm Optimization (PSO), as tracking method, along with a cost function calculated by a Linear Discriminant Functions (LDF) which utilizes RGB (Red, Green and Blue) histogram of image blocks for each particle to characterize the target object. Succinctly, the developed algorithm detects the target object through motion estimation and obtains the optical flow after the second captured frame. This process is performed for training of the cost function for the PSO method. As mentioned before the cost function is implemented using Linear Discriminant Functions, which are train based on RGB histograms of blocks about the current frame of each particle to characterize of the target object. The characteristics are divided into two classes target and non-target. Then, for the new frame the object will be tracked by the PSO method, which is based on three key elements: inertia, local best position and global best position. These elements are used to update the displacement of the swarm and consequently track the target. Updates to the global best position and local best position are defined through an evaluation of the proximity obtained between the current value and the centroid of a target class obtained during training. Through empirically test the following features of the proposed algorithm were identified: fast convergence, due to appropriate results obtained with few iterations; low computational cost, when compared to common deterministic methods, because it significantly reduces the amount of operations; an ability to treat occlusions which do not exceed one-quarter ($1/4$) of the resolution of the video frame and satisfactory object tracking capability in objects with arbitrary and abrupt movements.

Keywords: tracking, particle swarm optimization, linear discriminant functions, histograms rgb.

Sumário

1	Introdução	1
1.1	Descrição do Problema	1
1.2	Motivação	2
1.3	Objetivos	3
1.4	Metodologia	3
1.5	Contribuição do Trabalho	3
1.6	Estrutura do Trabalho	4
2	Rastreamento Visual	5
2.1	Vídeo Digital	5
2.2	Métodos de Rastreamento Visual	5
2.2.1	Métodos Determinísticos	7
2.2.2	Métodos Estocásticos	10
3	<i>Particle Swarm Optimization (PSO)</i>	14
3.1	Fundamentação do Algoritmo PSO Baseado em Metáfora	14
3.2	Aplicações Comuns do PSO	15
3.3	Definição do Algoritmo PSO	18
3.4	Rastreamento por PSO	20
4	<i>Linear Discriminant Functions (LDF)</i>	25
4.1	Trabalhos com Aplicação de Análise Linear Discriminante	27
4.2	Metodologia para Aplicar a Análise Discriminante	28
4.2.1	Primeiro Estágio	28
4.2.2	Segundo Estágio	28
4.2.3	Terceiro estágio	30
4.2.4	Quarto Estágio	30
4.2.5	Quinto Estágio	31
4.2.6	Sexto Estágio	33
4.3	Utilização do LDF	33
4.3.1	Processo de Treinamento	33
4.3.2	Classificador LDF	34
5	Sistemas de Rastreamento Baseado em PSO e LDF	35
5.1	Algoritmo Proposto	36
5.2	Divisão dos Quadros de Treinamento com os de Rastreamento	38
5.3	Estimação de Movimento e Fluxo Óptico	39

5.4	Detecção do Objeto Alvo	41
5.5	Treinamento da Função de Custo	41
5.6	PSO + LDF + Histograma RGB	43
6	Resultados	46
6.1	Configurações dos Testes	47
6.2	Resultados dos Testes com Sistema de Rastreamento	50
6.3	Testes de Desempenho e Complexidade	57
6.4	Testes de Rastreamento do Algoritmo Proposto	60
6.5	Teste de Rastreamento de Vídeo com Oclusão	63
7	Considerações Finais	65
A	Tabelas com os Resultados dos Testes com Sistema de Rastreamento	67
A.0.1	Vídeo 1	67
A.0.2	Vídeo 2	68
A.0.3	Vídeo 3	69
A.0.4	Vídeo 4	70
A.0.5	Vídeo 5	71
A.0.6	Vídeo 6	72
	Referências	74

Lista de Figuras

2.1	Representação hierárquica de um vídeo digital.	6
2.2	Representação do objeto alvo rastreado.	7
2.3	Exemplo de rastreamento utilizando métodos baseados em modelos.	8
2.4	Exemplo de rastreamento utilizando métodos baseados em aparência.	8
2.5	Exemplo de rastreamento utilizando métodos baseados em malha e contorno.	9
2.6	Algoritmo do método de rastreamento visual por estimativa de estados.	10
3.1	O esquema de deslocamento da partícula.	16
3.2	Ilustração do enxame de partículas em uma iteração inicial.	22
3.3	Ilustração do enxame de partículas em uma iteração posterior a inicial.	22
3.4	Ilustração do enxame de partículas na mudança de quadro após movimentação do objeto.	23
3.5	Ilustração do enxame de partículas durante a redistribuição das partículas na área de reinicialização.	24
4.1	Ilustração da sobreposição desejável e indesejável da Análise Discriminante Linear.	26
4.2	Processo contendo as atividades para aplicação da Análise Discriminante Linear.	29
4.3	Exemplo da análise de dois grupos através do cálculo da função discriminante.	32
5.1	Fluxograma do sistema proposto.	37
5.2	Esquema de utilização dos quadros de entrada do sistema proposto.	39
5.3	Ilustração do processo de estimação de movimentos.	40
5.4	Quadro 1 do exemplo para demonstração do processo de estimação de movimento.	41
5.5	Quadro 2 do exemplo para demonstração do processo de estimação de movimento.	41
5.6	Resultado da diferença dos quadros 1 e 2 do exemplo para demonstração do processo de estimação de movimento	42
5.7	Fluxo Óptico obtido com base na diferença dos quadros 1 e 2	42
5.8	Exemplo de máscara baseada em quadros aleatórios do Video1.avi.	43
5.9	Exemplo do vetor de características de entradas, classe alvo e não alvo.	43
5.10	Exemplo do vetor de características de entradas, classe alvo e não alvo, apresentada como gráfico.	44
6.1	Quadros que tipificam os vídeos utilizados os testes.	48
6.2	Quadros do rastreamento realizado no Video1.avi utilizando bloco 16×16	51
6.3	Quadros do rastreamento realizado no Video2.avi utilizando bloco 16×16	52
6.4	Quadros do rastreamento realizado no Video3.avi utilizando bloco 16×16	53
6.5	Quadros do rastreamento realizado no Video4.avi utilizando bloco 16×16	54

6.6	Quadros do rastreamento realizado no Video5.avi utilizando bloco 16×16 . . .	55
6.7	Quadros do rastreamento realizado no Video6.avi utilizando bloco 16×16 . . .	56
6.8	Porcentagem de sucesso no rastreamento em blocos 8×8 e 16×16	56
6.9	Gráfico que apresenta o desempenho de tempo em CPU das relações que obtiveram sucesso no rastreamento utilizando bloco 8×8 e 16×16	58
6.10	Porcentagem de rastreamento com sucesso no bloco 16×16 distribuído por número de partículas.	59
6.11	Porcentagem de rastreamento com sucesso no bloco 16×16 distribuído por número de iterações.	59
6.12	Gráfico que apresenta o comportamento de execução do estimador de movimentos e da proposta aqui apresentada (PSO + LDF + Histogramas RGB). . . .	61
6.13	Quadros do teste de rastreamento utilizando o algoritmo completo, onde no primeiro, terceiro e no penúltimo quadro é realizado o processo de estimação de movimentos.	62
6.14	Quadros do teste de rastreamento utilizando o algoritmo proposto com a estratégia para rastrear alvos com oclusão.	64

Lista de Tabelas

6.1	Características dos vídeos utilizados para os testes.	47
6.2	Identificação do primeiro quadro de cada vídeo que será rastreado nos testes do sistema proposto.	50
6.3	Identificação das relações de parâmetro dos testes.	50
6.4	Resultados dos testes com o Video1.avi para os determinados parâmetros no estimador de movimentos.	60
6.5	Resultados dos testes com o Video1.avi para os respectivos parâmetros no rastreador proposto.	60
6.6	Parâmetros adotados no teste com janela de busca ampliada gradativamente no algoritmo proposto no Video5.avi.	63
A.1	Identificação das relações de parâmetro dos testes.	67
A.2	Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo com dimensões de 8×8 - Video1.avi.	68
A.3	Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 16×16 para o Video1.avi.	68
A.4	Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 8×8 para o Video2.avi.	69
A.5	Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 16×16 para o Video2.avi.	69
A.6	Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 8×8 para o Video3.avi.	70
A.7	Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 16×16 para o Video3.avi.	70
A.8	Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 8×8 para o Video4.avi.	71
A.9	Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 16×16 para o Video4.avi.	71
A.10	Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 8×8 para o Video5.avi.	72
A.11	Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 16×16 para o Video5.avi.	72
A.12	Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 8×8 para o Video6.avi.	73
A.13	Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 16×16 para o Video6.avi.	73

Abreviaturas

AC	Alternating Current
CFA	Constriction Factor Approach
CMOS	Complementary Metal-Oxide-Semiconductor
FPGA	Field-Programmable Gate Array
FPS	Frames Per Seconds
HEVC	High Efficiency Video Coding
HSV	Hue, Saturation, Value
LDF	Linear Discriminant Functions
LDA	Linear Discriminant Analysis
MDA	Multiple Discriminant Analysis
NFL	Nearest Feature Line
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
R-LDA	Regularized LDA
RAM	Randon Access Memory
RGB	Red Green Blue
SSS	Small Simple Size
SSD	Sum Squared Differences

Capítulo 1

Introdução

A monitoração eletrônica baseada em vídeo digital tem se tornado chave para a eficiência de diversas atividades. Pois ele permite a identificação eficiente de anomalias no local monitorado, identificação de indivíduos suspeitos, esclarecimento de fatos ocorridos, entre outras atividades. Na esfera pública, os departamentos responsáveis pelo Monitoramento de Fluxo de Tráfego de veículos automotivos e as Secretarias de Segurança Pública vêm adotando novos sistemas de monitoração em vias urbanas muito movimentadas e locais com alto índice de violência. Para as empresas privadas de segurança física patrimonial, em que é comum a comercialização de serviços de monitoração de circuitos internos, um adequado sistema de monitoração eletrônica permite a redução de custos.

Apesar de possuir inúmeras vantagens, a utilização de monitoração eletrônica baseada em vídeo digitais ainda possui problemas a serem abordados. Dentre eles pode-se citar: a necessidade de recuperação e análise manual do vídeo quando um fato ocorreu e demorou a ser detectado; o desafio da qualidade do vídeo, pois, geralmente, usa-se a codificação em baixa resolução espacial para obter menor volume de dados, comprometendo assim, a qualidade do vídeo codificado; e a dificuldade de identificar em tempo real o que deve ser tratado para reduzir o volume de dados que deverá ser armazenado, já que em muitas situações o monitoramento deve ser contínuo, não sendo possível ignorar qualquer instante.

Em diversas circunstâncias, a proposta de se aumentar a complexidade na análise e no processamento do sinal de vídeo digital, de modo a requisitar maior consumo de recursos computacionais, pode ser justificada por uma redução significativa no armazenamento de dados de vídeos digitais. Portanto, este trabalho propõe um algoritmo de rastreamento que utiliza o método de Otimização por Enxame de Partículas (PSO - *Particle Swarm Optimization*) com uma função de custo baseada na Função Linear Discriminante (LDF - *Linear Discriminant Function*) e histograma RGB (*Red, Green and Blue*) para classificação de objeto alvo. Tal algoritmo visa oferecer a capacidade de identificação da trajetória de indivíduos em sistemas de monitoração eletrônica em ambiente restritos.

1.1 Descrição do Problema

O rastreamento da trajetória de objetos em meios digitais com qualidade e com baixo custo computacional é um desafio, devido a limitação de diversos recursos computacionais que realizam este tipo de processamento e a complexidade no reconhecimento de objetos de forma automatizada. A complexidade no tratamento da informação visual se dá por diversas influências, dentre

elas, é importante ressaltar as variações de luminância, sombras, texturas, posição, ângulo, deformação, escala e oclusões. A complexidade no tratamento é ampliada quando se tem uma sequência de informações visuais através de um vídeo, pois, estas aumentam muito quantidade de informações a serem tratadas, além de agregar novos problemas, os quais pode-se citar o rápido deslocamento dos objetos, deslocamento arbitrários, a deformação de objetos e oclusões dos objetos, impedindo o acompanhamento dos objetos, anteriormente identificados.

Uma forma de resolver os problemas apresentados em sequências de imagens de um vídeo é a integração de dois importantes mecanismos, a estimação de movimento, que permite a identificação de objetos em movimento através do cálculo da diferença entre dois quadros do vídeo, e o rastreamento visual, que obtém o deslocamento do alvo no decorrer dos próximos quadros do vídeo. A área de rastreamento visual tem demandado diversas pesquisas buscando a redução da complexidade computacional e a otimização da qualidade dos processos apresentados.

Em trabalhos anteriores [1], [2], [3] [4], a utilização do método de PSO para o rastreamento visual se demonstrou eficiente. Por diversas características, dentre elas: a rápida convergência, baixo custo computacional, capacidade de tratar pequenas oclusões, movimentos arbitrários e movimentos abruptos. Ainda em trabalhos anteriores [4], é apresentado um algoritmo hierárquico que utiliza o método de PSO para rastreamento baseado em movimentos do corpo humano. Tal algoritmo faz uso de um função de custo baseada na função Somatória das Diferenças Quadráticas (*Sum Squared Differences* - SSD) para alcançar o propósito apresentado anteriormente.

Já Zheng e Meng [3], que também utiliza o método de PSO para realizar o rastreamento, utiliza uma função de custo baseada em histogramas acumulativos, porém antes de se obter o histogramas é feita a transformação do espaço de cores RGB para HSV (*Hue, Saturation, Value*), a mudança dos espaços de cores de dá com o intuito de separar componentes estáveis de componentes ruidosos.

Hsu e Dai [1] implementou um algoritmo de rastreamento baseado no método de PSO para múltiplo alvos, buscando a utilização em aplicações de tempo real, em que foi utilizada a diferença entre histogramas em escalas de cinza para comparar cada partícula no espaço de busca com o objeto alvo e múltiplos enxames. Apesar do método criado demonstrar boa eficiência foi apresentada a necessidade de maior precisão nas buscas por características dos alvos que estão sendo rastreados.

1.2 Motivação

Atualmente, com o novo padrão de codificação de vídeo denominado *High Efficiency Video Coding* (HEVC) [5], que supera o H.264/AVC em termos de taxa-distorção, introduzindo novas técnicas de codificação, pode gerar maior complexidade de codificação. Logo, um algoritmo de rastreamento visual bem ajustado e de baixa complexidade capaz de identificar as regiões de interesse dentro de um sistema de monitoramento pode oferecer uma compactação diferenciada nesta região, permitindo um melhor desempenho do novo padrão de codificação. Além disto, um rastreamento de indivíduos em sistemas de monitoramento podem permitir realizar operações de aproximação (*zoom*) e movimentação da câmera de forma automática.

1.3 Objetivos

Este trabalho tem como objetivo geral a proposta de um algoritmo de rastreamento que utiliza o método de Otimização por Enxame de Partículas com uma função de custo calculada por uma Função Discriminante Linear, que utiliza histograma RGB dos blocos de cada partícula para caracterização do objeto alvo. Para desenvolver esta abordagem foi necessário o cumprimento dos seguintes objetivos específicos:

- Estruturação de uma função de custo para o algoritmo de rastreamento baseado no método de Otimização por Enxame de Partículas que ofereça a caracterização eficiente do alvo pela suas cores.
- Estruturação de um algoritmo que permita a junção do método determinístico para indicar a localização do alvo através de movimentos e treinar a função de custo.
- Identificação de parâmetros mais indicados para o algoritmo de rastreamento proposto, para caracterização do mesmo.
- Comparação do desempenho do algoritmo proposto com o método determinístico baseado na estimação de movimentos.

1.4 Metodologia

Para alcançar os objetivos apresentados, realizou-se uma extensa revisão bibliográfica em busca de trabalhos e materiais sobre os métodos classificação supervisionados, a otimização por enxame de partículas e o rastreamento visual para identificar os métodos satisfatórios para solução do problema.

A codificação do algoritmo de rastreamento foi realizada no programa Matlab. Tal escolha é devido a possibilidade de se utilizar dos diversos recursos oferecidos através de suas bibliotecas estáveis elaboradas para a área de processamento de imagens e visão computacional, de modo a simplificar o processo de codificação e garantir a estabilidade para a realização dos testes.

A comparação do algoritmo desenvolvido com o método determinístico baseado na estimação de movimentos, implementado com conceitos apresentados no trabalho de Richardson [6], avalia o desempenho de tempo na utilização da CPU, identificado através da ferramenta Matlab.

1.5 Contribuição do Trabalho

A principal contribuição do trabalho é a incorporação de uma técnica de classificação supervisionada, baseada em Funções Lineares Discriminante, ao modelo estocástico de rastreamento visual gerado a partir do método de Otimização por Enxame de Partículas. Tal técnica de classificação foi elaborada para fazer uso de histogramas RGB de blocos dos quadros para caracterizar o alvo, sendo incorporada na função de custo do método de Otimização por Enxame de Partículas. A implementação da proposta mostrou-se de grande valia prática, pois, requer muito menos recursos computacionais do que o método de rastreamento determinístico avaliado. E ele oferece boa resistência a movimentos abruptos e oclusões do objeto alvo.

1.6 Estrutura do Trabalho

A estrutura deste trabalho é composta por sete capítulos, que são apresentados a seguir resumidamente, e que formam o conjunto da obra:

O Capítulo 1, apresenta uma breve introdução do conteúdo que compõe o presente trabalho, sendo apresentada a descrição do problema, a motivação, os objetivos, a metodologia, a contribuição do trabalho e a estrutura do trabalho.

No Capítulo 2, são apresentados os conceitos teóricos acerca de rastreamento visual, em específico são tratados os conceitos de vídeo digital e as metodologias de rastreamento visual.

Os conceitos teóricos acerca da Otimização por Enxame de Partículas (PSO - *Particle Swarm Optimization*), em específico, a fundamentação do algoritmo com base em sua metáfora, as aplicações mais comuns do algoritmo, os detalhes do funcionamento e a adaptação para o rastreamento visual, são apresentados no Capítulo 3.

Os conceitos teóricos acerca de Função Discriminante Linear (LDF - *Linear Discriminant Functions*), desde as metodologias adotadas para a sua utilização do LDF até a aplicação em trabalhos anteriores, são apresentados no Capítulo 4.

Já o Capítulo 5, apresenta detalhadamente o sistema de rastreamento baseado no PSO e LDF, desde a obtenção dos quadros até a realização do rastreamento e perda do alvo.

As configurações dos testes e os resultados do algoritmo proposto são apresentadas no Capítulo 6.

Por último, no Capítulo 7 são apresentadas as principais conclusões acerca do trabalho desenvolvido, incluindo sugestões de possíveis trabalhos futuros com novas implementações.

Capítulo 2

Rastreamento Visual

Neste capítulo são apresentados os conceitos básicos de vídeo digital e rastreamento visual, além das principais técnicas consolidadas na área de rastreamento visual, que serviram de parâmetro de comparação teórica com o método de otimização de rastreamento visual PSO.

2.1 Vídeo Digital

Um vídeo digital é definido como uma representação do mundo real em cenas visuais, chamadas de quadros (imagens digitais) que são capturados por dispositivos que geram a representação de uma cena natural [6]. Tais quadros são repetidos, geralmente, em um curto espaço de tempo e em sequência, provocando a sensação natural do movimento. Logo, um vídeo digital é uma sequência de quadros (imagens digitais) que são organizados em um período contínuo. Cada quadro do vídeo é formado por um conjunto de *pixels* (abreviação de *picture elements*) que é o menor elemento da imagem, organizada em uma matriz [7], para que seja apresentado.

O quadro (imagem digital) que compõe um vídeo pode ser definido como uma função bidimensional $f(x, y)$; onde x e y identificam as coordenadas (localização) de cada *pixel* e os *pixels* são definidos através do valor de f no quadro [7]. O vídeo pode ser denotado por f_i , onde $i \in [0, \dots, \aleph]$, onde cada f_i representa um quadro.

Em sua estrutura básica o vídeo digital possuem três elementos: cenas, tomadas e quadros. A Figura 2.1 apresenta a representação hierárquica de um vídeo digital [8].

- Os quadros, visto anteriormente, são os elementos básico de um vídeo, ou seja, são as imagens capturadas ou geradas que permitem formar as informações de um vídeo [9].
- As tomadas são sequências de quadros organizados de forma contígua e representam uma ação contínua no tempo ou no espaço [10].
- As cenas são um conjunto de tomadas que representa uma sequência temporal de eventos naturais (fatos) comuns [11] [10] podendo ser classificadas como estáticas ou dinâmicas.

2.2 Métodos de Rastreamento Visual

O rastreamento visual é processo aplicado em vídeos (sequências de imagens) com o objetivo de localizar um ou mais objetos ou características desejadas [12], geralmente identificadas como

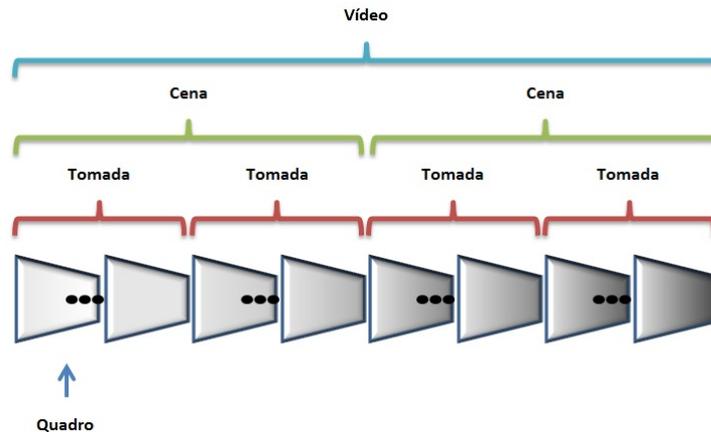


Figura 2.1: Representação hierárquica de um vídeo digital, que é composto por um conjunto de cenas, formadas por uma sequência de tomadas, que, por sua vez, são formadas por diversos quadros [8].

alvo. O rastreamento visual possui inspiração no processo de localização e acompanhamento, pelo globo ocular natural, de objetos em movimento. Em trabalhos prévios [13] [14] [15], é definido que em sua forma elementar o rastreamento compõe-se de um modelo de observação espacial, um modelo de representação do objeto rastreado e um algoritmo de rastreamento. Com base neste contexto as aplicações do rastreamento visual são diversas, dentre elas se destacam:

- A detecção automática de objetos através do movimento;
- A navegação de veículos não tripulados, possibilitando desvios de obstáculos e a identificação do caminho correto;
- A identificação de fluxo de tráfego nas estradas, através da contabilização de veículos;
- A monitoração com a finalidade de identificar atividades suspeitas ou inadequadas;
- Obtenção de informações relativas a gestos ou comportamentos humanos;
- Anotação automática de características dos vídeos pré-existente; e
- Controle em linhas de produção industrial.

Em [16], as formas comuns de identificação do objeto alvo que está sendo rastreado é através dos seguintes elementos: um ponto ou símbolo centroide sobre o alvo, diversos pontos sobre o alvo, retângulo sobre o alvo, elipse sobre o alvo, múltiplas elipses articuladas sobre o alvo, modelo em forma de esqueleto no alvo, contorno por pontos na borda do alvo, contorno através de linha nas bordas do alvo e preenchimento total do alvo em um quadro do vídeo. A Figura 2.2 apresenta as formas comuns de identificação do objeto alvo durante o rastreamento.

A seguir os métodos de identificação de objetos são organizados em determinísticos e estocásticos, tal classificação se dá buscando a segmentação de uma característica muito forte que tem surgido, que é baseada na busca por redução de custo computacional pelos métodos estocásticos.

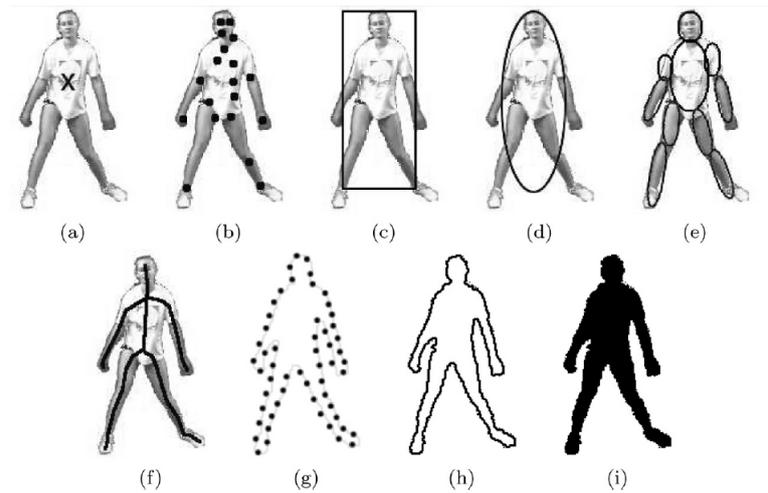


Figura 2.2: Representação do objeto alvo rastreado, são: (a) um ponto centroe sobre o alvo, (b) diversos pontos sobre o alvo, (c) retângulo sobre o alvo, (d) elipse sobre o alvo, (e) múltiplas elipses articuladas sobre o alvo, (f) modelo em forma de esqueleto no alvo, (g) contorno por pontos na borda do alvo, (h) contorno através de linha nas bordas do alvo e (i) o preenchimento total do alvo [16].

2.2.1 Métodos Determinísticos

Os métodos determinísticos de identificação dos objetos alvos rastreados em vídeos podem ser classificados em cinco grupos [17]: métodos baseados em modelos, métodos baseados em aparência, métodos baseados em contornos e malha, métodos baseados em características e métodos híbridos.

Os métodos baseados em modelos inicialmente exploram o formato comum dos objetos identificados na cena. Este método é sugerido em situações em que o objeto alvo está sem oclusão ou com oclusão parcial, pois desta forma não perderá o objeto alvo durante o rastreamento. O método possui algumas desvantagens, dentre elas: elevado custo computacional, necessita de todos os modelos geométricos bem detalhados antes de iniciar suas tarefas e incapacidade de generalização em caso de distorção. A Figura 2.3 apresenta um exemplo de rastreamento baseados em modelos.

Os métodos baseados em aparência realizam o rastreamento através da relação de regiões rudemente correspondente de um objeto no vídeo em sua forma bi-dimensional, possuindo como base um modelo dinâmico, para isto, geralmente, usam-se cores, texturas e movimento. Este método não é interessante em cenas que o objeto alvo possuem deformações complexas, pois pode-se impossibilitar o rastreamento do objeto alvo. A Figura 2.4 apresenta um exemplo de rastreamento baseados em aparência.

Os métodos baseados em contornos visam rastrear apenas as características do contorno do objeto sem analisar *pixels* internos do objeto, inicialmente este método depende do movimento para identificar o contorno e posteriormente é necessário se adaptar para continuar o rastreamento. Este método possui alta complexidade computacional e não é indicado em cenas que

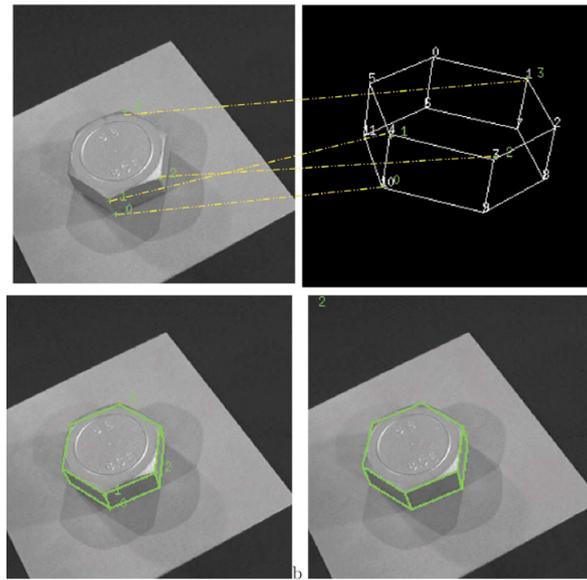


Figura 2.3: Exemplo de rastreamento utilizando métodos baseados em modelos, onde em (b) é criado um modelo geométrico a partir de (a) e em (c) e (d) tal modelo pôde ser identificado [18].

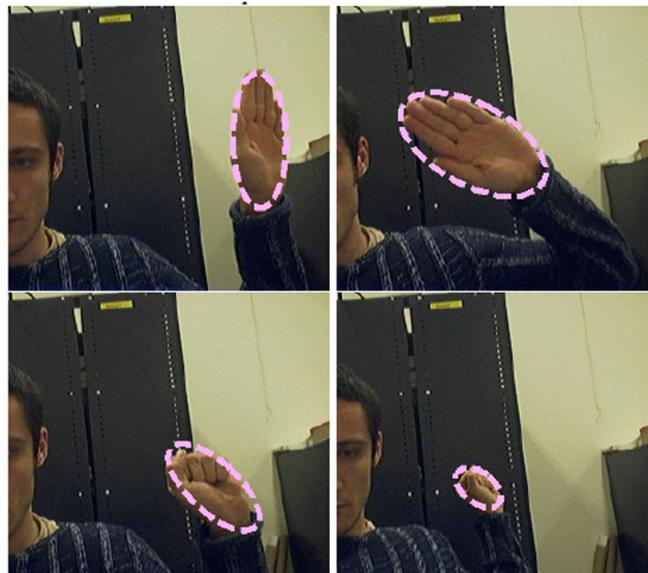


Figura 2.4: Exemplo de rastreamento utilizando métodos baseados em aparência, onde a cor da região da mão é rastreada independente da modificação do tamanho ou formato da forma [19].

possuem movimento bruscos com objetos alvo de corpo flexível (não rígido), pois, geralmente, o rastreador não consegue se adaptar rapidamente as mudanças do contorno. O método baseado em malha, que é composto por um conjunto de triângulos delimitados pelo contorno do objeto alvo, e outros métodos de modelos deformáveis têm sido adotados no lugar do método baseado em contornos, pois conseguem sanar a incapacidade de rastrear objetos alvo de corpo flexível (não rígido). Porém, tais métodos não conseguem solucionar problemas de oclusões. A Figura 2.5 apresenta um exemplo de rastreamento baseados em malha e em contorno.

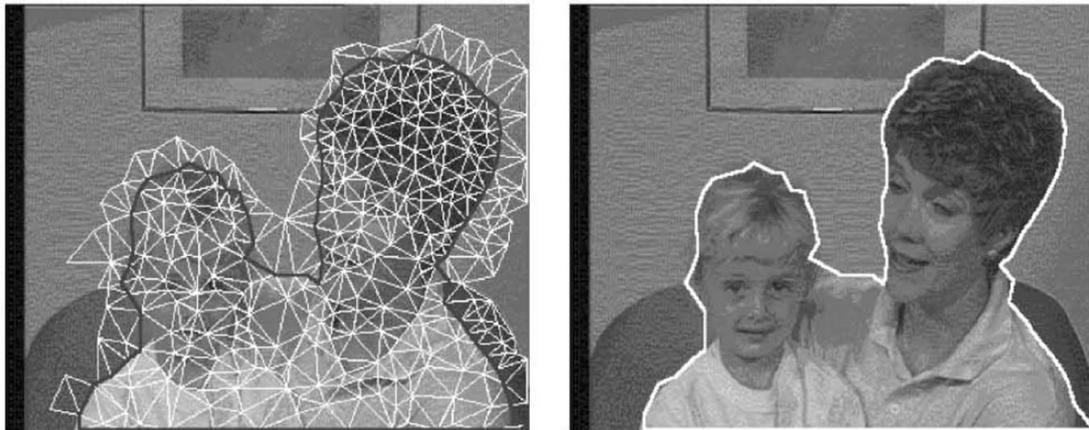


Figura 2.5: Exemplo de rastreamento utilizando métodos baseados em malha (do lado esquerdo) e contorno (do lado direito) [20].

Os métodos baseados em características utilizam as diversas características presentes em um objeto no vídeo para rastrear parte do objeto alvo. O rastreador pode, por exemplo, utilizar apenas os cantos (partes) que formam o objeto como estratégia de rastreamento. Tal método de rastreamento é estável e é sugerido em cena que o objeto alvo possui pequenas oclusões, isto sem impossibilitar o rastreamento do objeto alvo. Tal método não é sugerido quando na cena existem diversos objetos com partes semelhantes, pois o rastreador teria dificuldades de definir quais características pertencem a um determinado objeto e não a outro.

Já os métodos híbridos, unem o melhor de dois métodos, como é o caso do método baseado aparência com o método baseado em características, onde o objeto é definido como uma entidade para que, posteriormente, ela possa ter suas partes rastreadas. Esta estratégia possui dois níveis: o de objeto e o de região, em que o nível de objeto define a topologia do alvo e o nível de região define a topologia de áreas homogêneas que formam o alvo. Tal método oferece a capacidade de tratar deformações no objeto alvo. A desvantagem deste método é o aumento da complexidade que é devido a necessidade de projetar e adaptar as regiões de um quadro para outro.

2.2.2 Métodos Estocásticos

Existem diversos métodos de rastreamento visual que fazem uso de procedimentos estocásticos, nesta seção são abordados apenas alguns dos mais comuns, que serão agrupados em métodos por estimação de estados e métodos por associação de dados.

Métodos por Estimação de Estados

O processo de rastreamento visual por estimação de estado pode ser dividido em duas etapas: uma chamada de Predição de Estados, responsável pela obtenção do estado previsto baseado no quadro atual e o anterior; e outra chamada Atualização das Medições, responsável pela obtenção da nova estimativa através de uma medição do estado estimado anteriormente [21].

O Diagrama apresentado na Figura 2.6 apresenta a algoritmo do método de rastreamento visual por estimativa de estados.

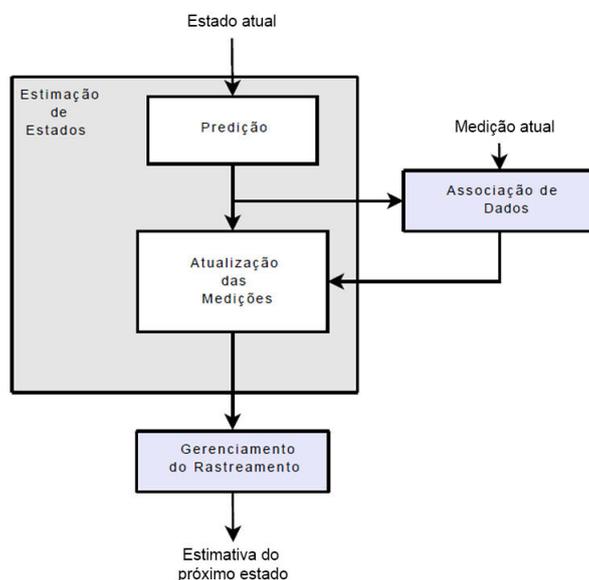


Figura 2.6: Algoritmo do método de rastreamento visual por estimativa de estados, que é dividida em duas etapas: a “Predição”, que é baseada no “Estado Atual”; e a “Atualização das Medições” que é baseada na interação da “Predição” com a “Medição Atual” que gera a “Associação de Dados” e que junto com a “Predição” permitirá a “Atualização das Medições”. Permitindo o “Gerenciamento do Rastreamento” e, por fim, a “Estimativa do Próximo Estado” [21].

Assim como apresentado na Figura 2.6, a estimação de estados é dividida em duas etapas, em que a “Predição” interage com o “Estado Atual” que permitirá a “Atualização das Medições” baseadas na interação da “Predição” e da “Medição Atual” que realiza a “Associação de Dados” e que junto a “Predição” permitirá a “Atualização das Medições”. Permitindo o “Gerenciamento do Rastreamento” e, por fim, a “Estimativa do Próximo Estado”. A escolha adequada do modelo de alvo é essencial para a eficiência deste método de rastreamento visual [21].

- Filtro de Kalman

O Filtro de Kalman consiste num conjunto de equações matemática consideradas computacionalmente eficientes para problemas de estimação de estado, pois minimiza o erro quadrático médio [22].

O Filtro de Kalman pode ser aplicado em sistemas com variáveis de estado contínuo, em que apresentam ruídos na medição. Dentre os sistemas comuns que utilizam o Filtro de Kalman, pode-se citar: trajetórias de aeronaves e mísseis, rastreamento de pessoas e objetos, reconstrução da trajetória de partículas, determinação de correntes oceânicas e acompanhamento acústico de submarinos [23].

O Filtro de Kalman pode ser utilizado em estimativas analíticas de problema em que o estado será estimado com base em um sistema com processo linear e modelos de medidas com incerteza gaussianas [23]. E a função de densidade de probabilidade, que é uma distribuição gaussiana inteiramente determinada por seu valor de média e matriz de covariância sobre um vetor de estados, define como será a transição de estados do processo.

- Filtro de Interação de Múltiplos Modelos

Em trabalhos anteriores [24] [25] [26], o Filtro de Interação de Múltiplos Modelos utiliza vários filtros paralelos interligados, em que cada filtro assume um dos modelos de modificação possíveis. Logo, em uma amostragem, entre dois quadros consecutivos, pressupõem-se que um dos modelos descreve a movimentação do alvo. Durante a execução todos os filtros são utilizados e a estimação do estado global é a combinação das estimativas dos filtros individuais.

O processo de rastreamento visual com o Filtro de Kalman, com um simples modelo de movimento, pode encontrar dificuldades na filtragem caso o cenário apresente objetos alvo com algumas manobras de mudança de trajetórias. Neste cenário, é indicado o uso de uma ferramenta que possua a capacidade de utilizar diversos modelos de modificação de trajetória ou mudanças probabilísticas dos modelos associados ao objeto alvo, que é o caso do Filtro de Interação de Múltiplos Modelos [27].

Rastreamento por Associação de Dados

A associação de dados consiste no problema de identificar quais características são inicialmente notadas. Cada medida individual é associada com características requeridas. Isto é, para que ao longo de toda a amostragem as medidas e características sejam coerentemente explanadas [28]. Ou seja, é a definição do processo de mensuração, para qualquer valor que foi utilizado no estimador de estado, onde é assumida a presença de conjuntos de rastreadores em cada amostragem e tais medidas são utilizadas na inicialização ou atualização dos rastreadores [21].

Os algoritmos relacionados ao rastreamento por associação, que serão apresentados são: Procedimento por Bloqueamento, Filtro de Vizinhança-próxima, Filtro de Associação Probabilística Conjunta de Dados e Filtro de Rastreamento por Múltiplas Hipóteses.

- Procedimento por Bloqueamento - *Gating Procedure*

Bloqueamento é um problema de ordem geral na associação de dados, determinando a probabilidade de associação significativa entre dois grandes conjuntos lineares, composto

por elementos de padrões multidimensionais [29]. Portanto, o Procedimento por Bloqueamento é adotado na seleção de medições candidatas para associação do rastreamento, permitindo a aceleração da atividade de associação de dados devido a redução da região de busca [21]. Isto, para medições relativas ao alvo de interesse no espaço de medição.

Já [30] aborda bloqueamento como Seleção de Medição (*Measurement Selection*) e define como a seleção de um subconjunto de medidas que, uma vez que o objeto alvo existe e é detectado, contém a detecção de objetos com alta probabilidade em cada amostragem. Esta probabilidade resultante denominada “Probabilidade de Bloqueamento” são, geralmente, calculadas como parte deste processo, isto, para fins de organização computacional.

- Filtro de Vizinhança-próxima - *Nearest Neighbor Filter*

O filtro de vizinhança-próxima é definido como uma medição, denotada $y_k(i)$, cuja a distância estatística do filtro de medição previsto é a menor entre todas as medições validadas y_k , que é considerada como sendo a medição alvo [30]. Este é alicerçado em cinco hipóteses recorrente e fortes:

1. É verdade que o objeto sempre existe e sempre é detectado.
2. A medida que está mais próxima (estatisticamente) para a medição é predito a partir do objeto.
3. Todas as outras medidas são de desordem (não alvo).
4. O movimento do objeto obedece estatísticas Gaussianas lineares.
5. O ruído de medição é Gaussiano branco.

O filtro de vizinhança-próxima é uma variação do método por bloqueamento (*Gating*) [31] e possui a probabilidade de associação determinada pela escolha da medição $y_k(i)$ a partir do conjunto de y_k , que possui as medições ordenadas através de sua distância estatística, que é determinada por um teste qui-quadrado [32]. Em algumas situações com alta desordem ou densidade de medições falsas, a medição de vizinhança-próxima pode obter resultados ruins. Isso por não considerar que a trajetória possa estar relacionada a um determinado tipo de alvo [30].

- Filtro de Associação Probabilística Conjunta de Dados - *Joint Probabilistic Data Association Filter*

A Associação Probabilística Conjunta de Dados é um método, de aproximação sub-ótima de simples verificação, que surgiu para solucionar a associação de dados em problemas de rastreamento de múltiplos alvos [32]. Suas associações entre as trajetórias conhecidas e as últimas observações são realizadas sequencialmente. E em cada amostra, são enumeradas as várias associações possíveis entre as observações e as trajetórias conhecidas, para que sejam realizados os cálculos de probabilidade de associação, através do conjunto β_{ik} , em que cada β_{ik} é a probabilidade da i -ésima observação do k -ésimo trajeto. Uma vez calculada a associação, o estado de um alvo é estimado através de um algoritmo de filtragem e este possível estado é ponderado pela probabilidade de associação. Por fim, o estado de um alvo é estimada pela soma de todos os possíveis estados ponderados [33].

Por utilizar uma medida de qualidade da trajetória pelo procedimento de discriminação de falso trajeto, a Associação Probabilística Conjunta de Dados assume que em cada trajeto existe apenas um único alvo a ser seguido. E que diferentes alvos (rastreados) podem seguir diferentes modelos de trajetórias, porém todas as tentativas dos diversos alvos (rastreados) são fundidas em uma única tentativa. E a atualização da trajetória se dá por meio de um estimador de associação probabilística de dados [30].

Existem dois tipos de modelos do Filtro de Associação Probabilística Conjunta de Dados um paramétrico, que assume a distribuição de Poisson (*Poisson distribution*) e um não-paramétrico, em que assume uma distribuição uniforme [34].

- Filtro de Rastreamento por Múltiplas Hipóteses

O Rastreamento por Múltiplas Hipóteses teve origem na principal dificuldades encontradas no rastreamento de múltiplos alvos o problema de associação de medições com trajetórias apropriadas, ainda mais quando se tem relato de perda, alvo desconhecido e desordem [35]. Com base neste contexto, Reid, em 1979, desenvolveu este método para calcular as probabilidades de várias hipóteses de associações de dados. Através deste método, foi possível a síntese de diversas outras características, dentre elas: a iniciação do rastreador, a correlação de múltiplas verificações, capacidade de processar conjuntos de dados com medições falsas ou inexistentes.

O algoritmo de Rastreamento por Múltiplas Hipóteses é recursivo e faz uso de medições posteriores para auxiliar em correlações anteriores. Em trabalhos anteriores [36], é apresentado que tal método mantém um banco de filtros de Kalman, onde cada filtro corresponde a uma hipótese específica sobre o alvo estabelecido. E cada hipótese correspondendo a uma associação postulada entre o alvo e uma característica medida. As múltiplas hipóteses surgem quando há duas ou mais características para os quais a associação correta é desconhecida. Estes métodos, no entanto, assumem que um conjunto de características discretas podem ser obtidas em cada intervalo de tempo, o que pressupõe que para tais existe um sensor. Por muitas vezes, isto não é verdade quando o rastreamento é de um objeto complexo.

Com base nas técnicas e filtros clássicos apresentados anteriormente, onde cada um possuía limitações específicas, é que tem surgido diversas técnicas de otimização do rastreamento visual, onde a maioria combinam as diversas técnicas, para se obter um bom desempenho. Uma técnica de otimização estocástica que tem se destacado recentemente é a baseada na Otimização por Enxame de Partículas (PSO - *Particle Swarm Optimization* [37] que consiste na criação de esquemas heurísticos com baixo custo computacional, permitindo a criação de sistemas de rastreamento em tempo real. Tal método será abordado no próximo capítulo.

Capítulo 3

Particle Swarm Optimization (PSO)

Neste capítulo é apresentado o estado-da-arte do método PSO (*Particle Swarm Optimization - Otimização por Enxame de Partículas*), sendo abordada a fundamentação do algoritmo baseada na metáfora de um enxame de abelhas, a definição geral do algoritmo PSO e, por fim, é apresentado o algoritmo PSO adaptado para o rastreamento visual e que serviu para a obtenção do algoritmo proposto no capítulo 5.

PSO é um método iterativo para otimização de funções contínuas não lineares, que foi modelado a partir da observação do comportamento social de algumas espécies de seres vivos, como enxame de abelhas, cardumes de peixes e bando de pássaros [38]. Na computação, este método é aplicado no campo da inteligência artificial bio-inspirada [39], assim como Algoritmos Genéticos [40] e Redes Neurais Artificiais [41]. Sua vantagem é a simplicidade de implementação, pois faz uso de operadores matemático primitivos, o baixo custo computacional, dada a relação velocidade e consumo de memória [38] e a simplicidade de compreensão.

PSO é um método estocástico que explora a inteligência coletiva [37], sendo composto por partículas, as quais representam integrantes de um grupo ou bando, que exploram uma área em busca de alimento, ou seja, exploram uma área para alcançar um determinado objetivo. A busca iterativa pela otimização deste processo é natural nos diversos contextos biológicos.

3.1 Fundamentação do Algoritmo PSO Baseado em Metáfora

O algoritmo PSO possui sua origem baseada na observação de sistemas biológicos. Portanto, nesta seção será apresentada a relação do algoritmo PSO com o enxames de abelhas. Naturalmente, o algoritmo matemático não é idêntico aos sistemas que inspiraram a sua criação [38], logo são apresentadas as divergências sob a ótica do autor do algoritmo, demonstrado em [42].

Algumas abelhas possuem a tarefa de colher o néctar e o pólen das flores [43]. Esta tarefa é modelada no PSO através dos indivíduos (partículas) do enxame, as flores são as regiões de interesse e o alvo é o néctar e o pólen. No PSO, a região de interesse pode ser definido como, pelo menos, um local ótimo com base em uma função aplicada em um espaço de busca. Tal função pode ser uma fórmula matemática, um algoritmo, ou ainda, o resultado de um processo, real ou simulado. O essencial é que seja possível calcular este valor em cada indivíduo do enxame.

As informações de onde se encontra uma determinada região de interesse só é trocada entre as abelhas, geralmente, quando as mesmas retornam para a colmeia [42]. Tal característica não foi modelada de modo semelhante, pois reduziria o desempenho matemático do algoritmo.

Na modelagem do PSO, o indivíduo possui a capacidade de comunicação de onde ele estiver diferenciando-o das abelhas.

O tamanho de um enxame de abelhas é aleatório [43], diferenciando do modelo adotado no PSO, que possui um número fixo de indivíduos antes de iniciar as suas tarefas. Já a distribuição das partículas do PSO no espaço de busca é modelado a partir das características de um enxame de abelhas. Como de início elas não têm certeza de onde encontrarão o néctar ou o pólen, elas se distribuem aumentando o campo de busca.

Assim como no enxame biológico, cada indivíduo guarda o caminho para localizar o alvo, ou seja, a taxa de deslocamento, obtida através do deslocamento do ponto de partida ao ponto aleatório na área de busca.

Como no enxame de abelhas, não é possível a comunicação remota [43], as características de comunicação se diferencia muito da modelada no algoritmo PSO, em que cada partícula deve receber algumas partículas informantes, possibilitando a identificação da partícula que encontrou um local ótimo, dentre todas as partículas do enxame.

Na modelagem, quanto maior a quantidade de informações armazenadas pela partícula, pior o desempenho e mais complexo o funcionamento. Porém, uma informação relevante é o melhor desempenho geral daquela partícula, a sua melhor posição vivida e a qualidade desta posição, perante a busca por seus objetivos.

Calcular a velocidade do próximo deslocamento na modelagem do PSO foi um dos desafios encontrados. Foi relevante considerar no processo que a própria partícula já carregava a informação do seu próprio desempenho, uma vez que elas foram distribuídas aleatoriamente, sendo necessário armazenar apenas a informação da partícula com o melhor desempenho.

Logo, na modelagem do PSO, os elementos essenciais combinados para permitir a realização do cálculo de velocidade, são: a própria velocidade, o próprio melhor desempenho, e o melhor desempenho entre os informantes.

A Figura 3.1 apresenta como funciona a relação entre os três elementos essenciais, a velocidade da partícula, o melhor desempenho da partícula, e o melhor desempenho entre as partículas, para calcular o deslocamento de cada partícula.

Assim como apresentado na Figura 3.1, tem-se que o deslocamento é baseado na velocidade atual da partícula, na direção ao seu melhor desempenho da partícula e a melhor partícula dentre todas as outras, sendo necessária uma ponderação linear, ajustada com coeficientes de confiança.

3.2 Aplicações Comuns do PSO

São diversas as aplicações que fazem uso do método de PSO, em [44] são sintetizadas algumas das diversas aplicações do PSO:

- **Análise de área para projeto de antenas:** com trabalhos aplicados na área de redes e arranjos de antenas, modelagem e projeto de antenas para banda larga, modelagem e projeto de arranjos Yagi-Uda, correção de falhas em arranjos, projeto de antenas multi-bandas, otimização de arranjos de antenas refletoras, padrões de reconstrução de radiação, antenas de arranjo adaptativo, dentre outros, projeto de antenas implantáveis e etc.
- **Análises médicas, aplicações biológicas e farmacêuticas:** aplicado em trabalhos que analisam tremores em pessoas com mal de Parkinson, inferência da rede de genes re-

- **Esquemas de distribuição de redes elétrica:** envolve planejamento de redes de transmissão, reconfiguração e expansão de redes, geração e distribuição, regulação de voltagem, gerenciamento de congestão e micro-grades.
- **Sistemas eletrônicos e eletromagnéticos:** envolve temas como o controle de sistema de transmissão AC, indutores em chips, células de combustível, controle de temperatura baseado em FPGA (*Field-Programmable Gate Array*), projeto de forma eletromagnética, filtros de micro-ondas, CMOS (*Complementary Metal-Oxide-Semiconductor*) de alta velocidade, Síntese de antena de arranjo linear, condutores, projeto genérico de eletromagnetismo e aplicativos de otimização, síntese de circuitos, projeto de circuitos digitais, configuração de FPGAs e processadores paralelos de arranjos.
- **Otimização de motores mecânicos e elétricos:** agrupa estudos em mecanismos de classificação de dados, controle de torque em locomotivas, controle de motor em veículos elétricos e híbridos, controle de velocidade em motor de indução, otimização de motores de combustão interna e otimização de sistemas de propulsão elétrica nuclear.
- **Geração de músicas e jogos para o mercado de entretenimento:** incluindo análise de nós folhas em árvores de jogos, reiteração do dilema do preso da Teoria dos Jogos, aprendizado em jogos de tabuleiro e *pong* solo, todos na área de jogos. Já na área de música, pode-se destacar o uso das técnicas de *swarms* para improvisar a composição interativa de música.
- **Detecção ou diagnóstico de falhas e sua recuperação:** estão envolvidos nesse tópico o diagnóstico de falhas de geradores de turbina a vapor, circuitos que recuperam-se automaticamente de falhas de componentes, classificação automática de defeitos em pastilhas semicondutoras, diagnóstico de falhas em circuitos digitais, isolamento defensivo de sistema de energia, serviço de restauração de distribuição de energia, diagnóstico de falhas de motores, dentre outros.
- **Análise econômica e financeira:** estão nesse grupo, sistemas de análise de risco, opção de precificação, seleção de carteira de investimento e mercado de energia elétrica.
- **Sistemas de predição e previsão:** envolve sistemas de predição da qualidade da água, previsão de cenários caóticos, previsão de carga elétrica, previsão de migração de elefantes, previsão de vazão e sistemas de previsão de tráfego urbano.
- **Segurança e aplicações militares:** estão inclusos redes de segurança, detecção de intrusão, criptografia e criptoanálise, geração de energia com restrições de segurança, identificação dos limites de segurança em sistemas de potência e otimização da eficácia de mísseis.
- **Computação gráfica:** são exemplos de aplicações, a apresentação gráfica de redes, redução da dimensionalidade, detecção de colisão em modelos gráficos, síntese de textura, multidões de partículas interativas e gráficos 3D.
- **Robótica:** são exemplos, controle de braços e manipuladores robóticos, corrida de robôs, busca robótica coletiva, aprendizado robótico sem supervisão, navegação não-tripulada de veículos, futebol de robô, planejamento de trajetória, localização de fontes de odores, mapeamento de ambiente, desvio de obstáculos e controle de voz de robôs.

- **Análise de imagens e vídeos:** possuem destaque pela quantidade de trabalhos que aplicam o PSO, destacando-se o reconhecimento de íris, detecção e reconhecimento de face, segmentação de imagens, imagens de radar de abertura sintética, classificação de imagens, inversão de medidas de reflectância de cores do oceano, fusão de imagens, classificação de *pixels*, detecção humana em imagens em infra-vermelho, detecção de defeitos, recuperação de imagens, imagens de microondas, síntese de textura, realce de contraste, detecção do sinal de parada no tráfego, planejamento de tratamento com imagens de raio-x em ortodontia, recuperação 3D com matriz de feixe estruturado, corte automático para fotografias digitais, reconhecimento de caracteres, dentre outras aplicações são exemplos de aplicações em imagens. No caso dos vídeos destacam-se otimização do MPEG, estimativa de movimento, rastreamento de objetos, controle de postura corporal e a detecção de incidentes de tráfego.

3.3 Definição do Algoritmo PSO

Antes de abordar o algoritmo PSO completamente faz-se necessário detalhar alguns termos, entre eles:

- **Objetivo:** é a região de interesse das partícula, ou seja, o alvo do enxame.
- **Partículas:** São indivíduos (agentes) que compõe o enxame e carregam consigo informações, localização e o valor (custo) de sua localização atual. E são identificadas por x .
- **Enxame:** É o conjunto formado por todas as partículas. Uma característica forte do enxame é que todas as partículas possuem um objetivo comum.
- **Espaço de Busca:** É a região de atuação do enxame, é um delimitador de local de distribuição de todas as partículas pertencentes ao enxame. Tal espaço de busca é identificado por d ao longo do trabalho.
- **Posição:** são as coordenadas n-dimensional de uma partícula que compõe o enxame, tal coordenadas identificam a localização da partícula no plano. Em um plano bi-dimensional tais coordenadas podem ser apresentadas através das variáveis x' e y' .
- **Função de Custo:** Também identificada como aptidão, ela representa o valor (custo) da localização de cada partícula, ou seja, identifica o quão próxima a partícula está do seu objetivo. Tal função deverá possui um único valor a ser armazenado na partícula. As funções de custo da partícula atual, melhor posição local e melhor posição global, foram identificada no trabalho como: J , J_p e J_g , respectivamente.
- **Melhor Posição Local:** Representa uma memória da partícula com as coordenadas e aptidão de onde ela esteve mais próxima de alcançar ou alcançou o seu objetivo. É uma variável que guarda informações da melhor função de custo encontrada pela partícula. A Melhor Posição Local será identificada como p ao longo do trabalho.
- **Melhor Posição Global:** Representa uma memória relacionada a todo enxame com as coordenadas e aptidão da partícula que esteve mais próxima de alcançar ou alcançou o objetivo. É uma variável que guarda informações da melhor função de custo encontrada dentre todas as partículas. A Melhor Posição Global será identificada como g ao longo do trabalho.

- **Velocidade:** é a movimentação de cada partícula no plano. Tal velocidade é obtida para cada partícula baseada no cálculo da inércia (velocidade que a partícula possuía) com a melhor posição local da partícula e a melhor posição global do enxame. A velocidade será identificada como v ao longo do trabalho.

Em trabalhos anteriores [45] [46], o algoritmo base do PSO é estruturado da seguinte forma, há um número finito total de partículas n , tais partículas podem se espalhar por um determinado espaço de busca d , a posição da i -ésima partícula na k -ésima iteração é representada por $x_k[i]$, a velocidade da partícula é representada por $v_k[i]$ e o valor de custo da partícula é $J_k[i]$.

Além dos elementos apresentados, têm-se a melhor posição da i -ésima partícula, ou seja, a mais próxima do alvo, nas iterações passadas e seu valor de custo associado, são representados por $p_k[i]$ e $J_{pk}[i]$, respectivamente e a melhor posição global dentre todas as partículas na iteração e o valor de custo representados por g_k e J_{gk} , respectivamente.

Para alcançar o objetivo em um número n de iterações o algoritmo PSO deve possuir 5 passos, são eles:

1. Gerar aleatoriamente numa iteração inicial as partículas, $x_0[i]$, em que $i \in [1, 2, \dots, n]$ e x_0 são as posições iniciais de todas as partículas do enxame.
2. Comparar as funções de custo da iteração passada com a da iteração atual em todas as partículas, sendo determinante para atualização da melhor posição da partícula $p_k[i]$.

$$p_k[i] = x_k[i], \text{ se } J_k[i] > J_{pk}[i] \quad (3.1)$$

3. Comparar as funções de custo da melhor posição global com a da iteração atual, de forma de determinar g_k .

$$g_k = x_k[i], \text{ se } J_k[i] > J_{gk} \quad (3.2)$$

4. Mudar a iteração através das Equações 3.3 e 3.4 apresentadas a seguir.

$$v_k[i] = K \cdot \{v_{k-1}[i] + c_1 \cdot \Gamma(g_k - x_k[i]) + c_2 \cdot \Lambda(p_k[i] - x_k[i])\} \quad (3.3)$$

$$x_{k+1}[i] = x_k[i] + v_k[i] \quad (3.4)$$

onde, c_1 e c_2 e K são coeficientes de ponderação. Γ , Λ são denotados por:

$$\Gamma = [\gamma_1, \gamma_2, \dots, \gamma_d].$$

$$\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_d].$$

onde $\gamma_i \in [0,1]$, $\lambda_i \in [0,1]$ ($i = 1, 2, \dots, d$) são números pseudo-aleatórios com distribuições uniformes.

5. Atualizar a iteração k para $k+1$ e repetir os passos de 2 à 5, até atingir o número máximo de iterações ou o algoritmo convergir (a mudança na velocidade das partículas ser insignificante).

No PSO alguns parâmetros podem aumentar ou reduzir o desempenho em termos de convergir para alcançar a posição desejada. Uma abordagem interessante para aumentar o desempenho

é a CFA (*Constriction Factor Approach* - Abordagem por Fator de Constrição) [46], que pode ser obtido através da Equação 3.5 apresentada a seguir.

$$K = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \quad (3.5)$$

Onde $\phi = c_1 + c_2$, $\phi > 4$. O comportamento dos enxames convergem gradualmente porque qualquer ϕ satisfaz $K < 1$ e os coeficientes de peso que se relacionam com a x_k , p_k e g_k são ajustados por apenas um parâmetro.

3.4 Rastreamento por PSO

Assim como mencionado no início deste capítulo, o PSO é muito aplicado em processos envolvendo tratamento de imagens e vídeos, além de ser utilizado em aplicações de visão robô. Nelas, uma aplicação específica que se destaca é o rastreamento de objetos. Assim, será apresentada a base teórica da adaptação do algoritmo PSO para atender as aplicações de rastreamento de objetos.

O algoritmo de PSO original é adequado para encontrar um objeto numa determinada imagem. Porém, para se trabalhar com rastreamento de objetos em vídeo digital, o PSO deve ser modificado devido a característica de movimentação do objeto.

No algoritmo original de PSO, no final da última iteração, as partículas convergem para uma determinada posição, um problema no rastreamento visual é que se no próximo quadro o objeto se movimentou de tal forma que saiu da área onde encontram-se todas as partículas o algoritmo perdeu o alvo.

Em trabalhos anteriores [45], foi proposta uma atualização periódica das variáveis $p_k[i]$ e g_k para evitar a perda do alvo. Os autores assumiram que somente um objeto está sendo rastreado e que esse objeto não se movimenta abruptamente entre quadros contíguos do vídeo.

Quando um novo quadro é analisado g_k , J_{gk} , $p_k[i]$, $J_{pk}[i]$ e $v_k[i]$ são armazenados e, em seguida, as partículas são distribuídas ao redor de g_k , antes da próxima iteração, de forma aleatória e dentro de um espaço de busca.

A seguir, g_k é determinado a partir das novas posições das partículas. Outro problema, é que ao convergir o algoritmo num determinado quadro chega em um ponto que as funções de custo J_{gk} e $J_{pk}[i]$ tem um valor alto. Quando objeto se mover no próximo quadro as partículas do enxame têm que buscar um valor de J_{gk} mais alto ainda, o que pode não ser possível. Logo, para um rastreamento estável, sempre que o quadro muda deve-se atualizar as variáveis $p_k[i]$ e g_k , e utilizar um fator de atenuação, identificados no algoritmo como α e β , respectivamente, para $J_{pk}[i]$ e J_{gk} .

Com base nas mudanças apresentadas é que se chegou ao algoritmo PSO adaptado para rastreamento visual que é apresentado a seguir. Onde para alcançar o objetivo em um número k de iterações. O algoritmo PSO deverá possuir 8 passos, são eles:

1. Gerar aleatoriamente numa iteração inicial ($k = 0$) a distribuição das partículas, $x_0[i]$, em que $i \in [1, 2, \dots, n]$, dentro de um espaço de busca.
2. Capturar um quadro (*frame*) do vídeo.

3. Comparar as funções de custo da iteração passada com a da iteração atual em todas as partículas, sendo determinante para atualização da melhor posição da partícula $p_k[i]$.

$$p_k[i] = x_k[i], \text{ se } J_k[i] > \alpha \cdot J_{p_k}[i], \text{ onde } 0 < \alpha < 1. \quad (3.6)$$

4. Comparar as funções de custo da melhor posição global com a da iteração atual, de forma de determinar g_k .

$$g_k = x_k[i], \text{ se } J_k[i] > \beta \cdot J_{g_k}, \text{ onde } 0 < \beta < 1. \quad (3.7)$$

5. Atualizar a iteração do enxame através das equações 3.8 e 3.9 apresentadas a seguir.

$$v_k[i] = K \cdot \{v_{k-1}[i] + c_1 \cdot \Gamma(g_k - x_k[i]) + c_2 \cdot \Lambda(p_k[i] - x_k[i])\} \quad (3.8)$$

$$x_{k+1}[i] = x_k[i] + v_k[i] \quad (3.9)$$

onde, c_1 e c_2 e K são coeficientes de ponderação. Γ , Λ são denotados por:

$$\Gamma = [\gamma_1, \gamma_2, \dots, \gamma_d].$$

$$\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_d].$$

onde $\gamma_i \in [0,1]$, $\lambda_i \in [0,1]$ ($i = 1, 2, \dots, d$) são números pseudo-aleatórios uniformes.

6. Retornar todos os indivíduos que saíram do espaço de busca em uma posição aleatória.
7. Realizar a iteração dos passos 3 - 6 até que o número máximo de iterações de um quadro (*frame*) seja atingido.
8. Reinicializar o enxame baseado no na fórmula 3.10 e retornar para o passo 2.

$$x_k[i] = g_k - \delta(R_{max} - R_{min})(2\gamma - 1). \quad (3.10)$$

Onde, R_{max} e R_{min} são os limites máximo e mínimo do espaço de busca, δ representa o limite de atribuição e $\gamma \in [0,1]$ é um número uniforme do pseudo-limite.

Portanto, o algoritmo do PSO para rastreamento deve funcionar, em uma iteração inicial $k = 0$, distribuindo todas as partículas sobre o espaço de busca, onde será eleita uma partícula melhor global g_0 após a análise da função de custo de todas as partículas, segundo os passos de 1 a 4 do algoritmo PSO apresentado anteriormente. A Figura 3.2 ilustra a iteração inicial do algoritmo PSO.

Nas próximas iterações, todas as demais partículas irão convergir e vão se direcionar para a partícula com melhor posição global g_k , aproximando-se do objeto alvo, passos de 5 a 7 do algoritmo PSO adaptado e apresentado anteriormente. A Figura 3.3 ilustra o momento em que as partículas estão convergindo, ou seja, com todas as partículas indo em direção a partícula com melhor posição global g_k .

Na mudança de quadro do vídeo, após um movimentação do objeto, todas as partículas podem perder o objeto alvo, inclusive a partícula com melhor posição global da interação atual g_k , assim como é apresentado na Figura 3.4.

Uma vez que o objeto alvo é perdido, faz-se necessária uma nova distribuição das partículas dentro da área de reinicialização, que está dentro da área de busca, isto segundo o passo 8 e

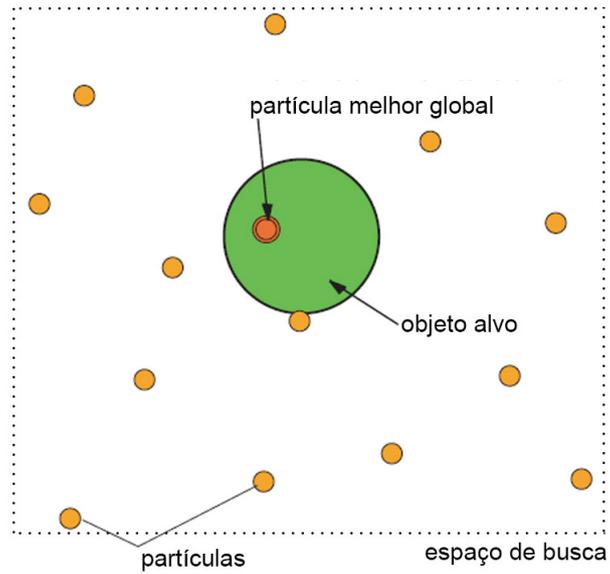


Figura 3.2: Ilustração do enxame de partículas em uma iteração inicial, $k = 0$, onde a melhor global é a que estiver sobre o alvo [45].

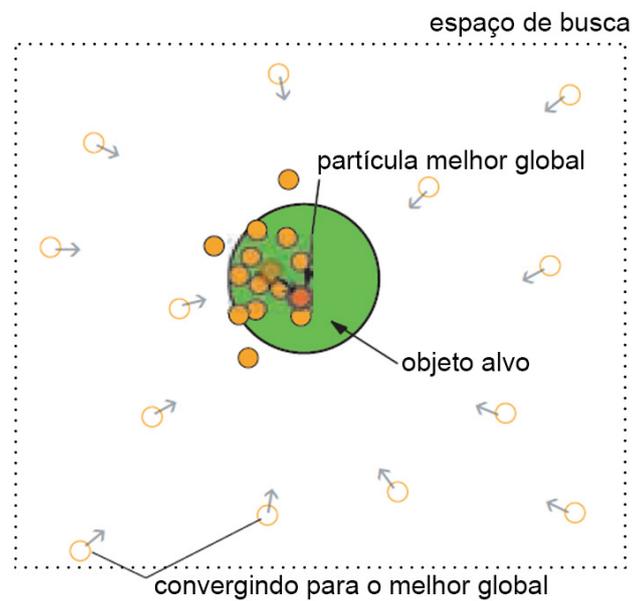


Figura 3.3: Ilustração do enxame de partículas em uma iteração posterior a inicial, onde as partículas convergiram para a melhor global [45].

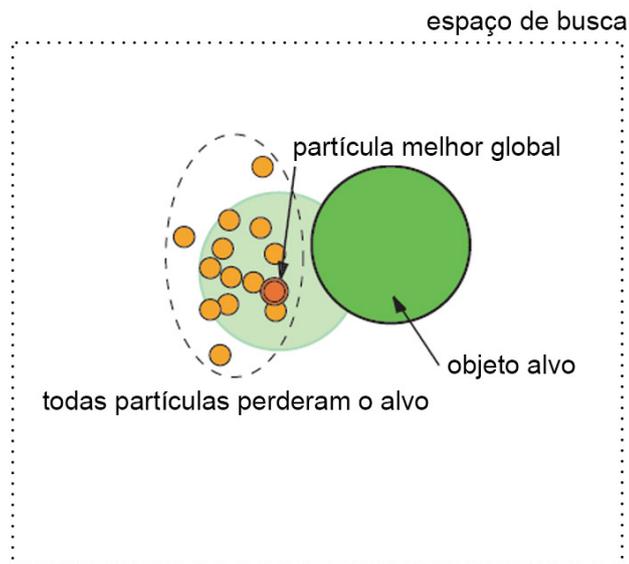


Figura 3.4: Ilustração do enxame de partículas na mudança de quadro após movimentação do objeto [45].

Equação 3.10 do algoritmo PSO adaptado e apresentado anteriormente. De modo a permitir a continuidade do rastreamento adequado do objeto alvo, assim como é apresentado na Figura 3.5.

O próximo capítulo aborda os conceitos gerais da Função Discriminante Linear (LDF - *Linear Discriminant Functions*), que é um classificador linear baseado em métodos estatísticos, que pode ser usado no PSO para analisar a função de custo de uma partícula e definir o quão próximo a partícula está do alvo.

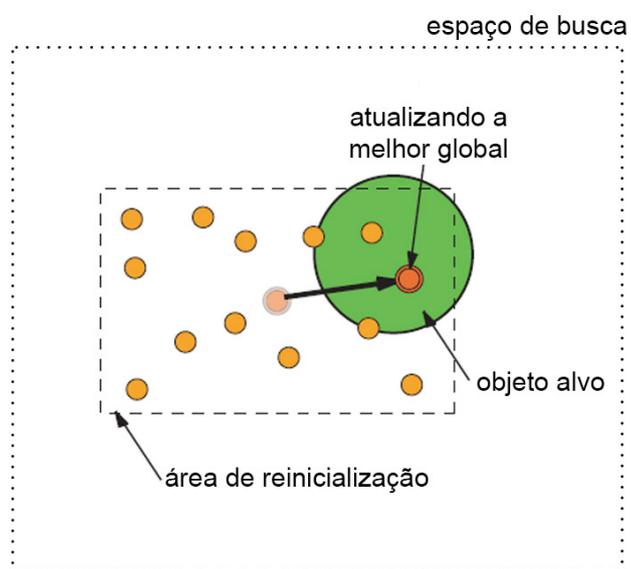


Figura 3.5: Ilustração do enxame de partículas durante a redistribuição das partículas na área de reinicialização [45].

Capítulo 4

Linear Discriminant Functions (LDF)

Neste capítulo são apresentados os conceitos gerais relacionado a Função Discriminante Linear (LDF - *Linear Discriminant Functions*), dividido em, metodologia para aplicar a análise discriminante, processo de treinamento e classificação do LDF na sua utilização. Tais conceitos fundamentam a elaboração da função de custo do algoritmo de rastreamento proposto baseado no método de PSO, além de definir a dinâmica de treinamento da função de custo do algoritmo proposto.

Fisher [47] introduziu o conceito “*discriminação*” ou “*separação*” no qual é abordado neste trabalho como um problema de classificação. Para ele, as características especiais de uma população permitem a identificação da mesma em um grupo. Estas características podem ser melhor analisadas através de certas funções lineares de medição.

A Função Discriminante Linear e a Discriminante Linear de Fisher são usadas para o processo de medição das características dos objetos, (x_1, \dots, x_n) , às quais são atribuídas as funções lineares de forma que populações (elementos, objetos) sejam melhores discriminadas [47]. Em trabalhos anteriores [48], o propósito da classificação ou análise discriminante é o de assimilar objetos para um dos diversos grupos ou classes (K) baseado em um conjunto de medidas $X = \{x_1, x_2, \dots, x_n\}$ obtidas de cada objeto ou observação.

Uma tarefa de análise discriminante tradicional é aquela utilizada em algumas situações onde, a partir de uma classificação prévia de um grupo alvo de objetos, baseado em um conjunto de variáveis $X = \{x_1, x_2, \dots, x_n\}$, deseja-se encontrar uma regra de classificação quantitativa de forma a prever a adesão de um dado objeto a uma das classes pré-determinadas [49], [50].

A Análise Linear Discriminante (LDA – *Linear Discriminat Analysis*) é um método que permite encontrar a pontuação para classificar objetos [49]. Tais pontuações são expressas na forma de combinações lineares de variáveis explanatórias que separam os grupos previamente definidos da melhor forma possível. A análise discriminante é uma técnica estatística apropriada para testar a hipótese de que a média do grupo de um conjunto de variáveis independentes para dois ou mais grupos são iguais [51].

A média das pontuações discriminantes de todos os indivíduos (objetos) dentro de um grupo leva a obtenção da média do grupo, também denominado de centróide. Cada grupo formado na análise possui um centróide que é um ponto de referência na formação deste grupo, ou seja, cada um dos elementos está dentro do raio de abrangência do grupo, possuindo desta forma uma distância mínima em relação ao centróide. Nesse sentido, se a sobreposição da pontuação é pequena, a função separa bem os grupos, porém, se houver grande número de sobreposição torna-se difícil separar os elementos e a classificação retorna resultados ruins na distribuição. A Fi-

gura 4.1 apresenta na parte superior a sobreposição desejável e na parte inferior a sobreposição indesejável.

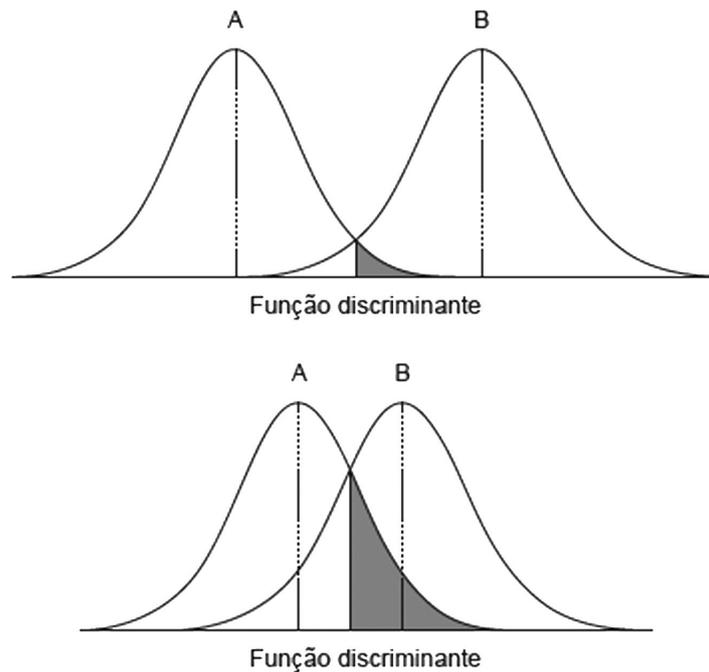


Figura 4.1: Ilustração da sobreposição desejável (na parte superior da figura) e indesejável (na parte inferior da figura) da Análise Discriminante Linear [51].

A análise discriminante é uma técnica adequada a problemas que envolvem uma variável independente e uma variável categórica (nominal ou não métrica), sendo que as variáveis independentes são variáveis métricas. As variáveis categóricas são responsáveis por nomear os grupos nos quais as variáveis métricas irão ajudar a classificar. Quando se possui duas classificações envolvidas, a técnica é referida como análise discriminante com dois grupos, e no caso de mais classificações envolvidas, a técnica é denominada análise discriminante multivalorada (MDA - *Multiple Discriminant Analysis*).

A técnica de análise discriminante envolve derivar uma variável (variável discriminante), que é a combinação de duas ou mais variáveis independentes que classificam cada um dos objetos nos grupos previamente definidos. A técnica é também denominada de Função Linear Discriminante, a qual faz a discriminação de cada um dos objetos através de uma função objetivo que busca maximizar as diferenças entre os grupos [51], com a diferença obtida da variância entre os grupos em relação a variância dentro do grupo. A Equação 4.1 representa a função discriminante linear.

$$Z_{jk} = a + W_1X_{1K} + W_2X_{2K} + \dots + W_nX_{nK} \quad (4.1)$$

Onde,

Z_{jk} = é a pontuação discriminante Z da função discriminante j para o objeto k ;

a = interceptação;

W_i = peso discriminante da variável independente i ;

X_{iK} = variável independente i do objeto k .

Conforme observado na Equação 4.1, a pontuação discriminante para cada objeto k na análise é uma soma de valores obtidos pela multiplicação de cada variável independente (X_{nK}) pelo seu peso discriminante (W_n). Através da Equação, pode-se observar, ainda, que na análise discriminante podem ser utilizadas várias funções discriminantes para obter a pontuação (Z_j), característica que possibilita diferentes pontuações para um dado objeto.

4.1 Trabalhos com Aplicação de Análise Linear Discriminante

A Análise Discriminante auxilia no processo de classificação dos objetos e, portanto, têm sido utilizada em pesquisas que envolve reconhecimento de padrões e reconstrução de imagens. Nesta seção serão apresentados alguns trabalhos relatados na literatura que utilizam essa técnica de forma a embasar a sua utilização aplicada a área de processamento de imagens digitais.

Martínez e Kak [52] apresenta um comparativo entre as técnicas de LDA e PCA (*Principal Components Analysis*) no contexto do paradigma baseado em aparência para reconhecimento de objetos, ambas bastante utilizadas nesse contexto. Apesar de mostrar em seu trabalho que o PCA pode apresentar em algumas circunstâncias resultados superiores ao da LDA, os autores reforçam que a LDA na grade maioria dos casos apresenta melhores resultados, principalmente quando envolve um grande conjunto de dados para treinamento. Os autores compararam as técnicas para identificação de faces, que atuam em um espaço $n \times m$ -dimensional, independente de questões relacionadas com localização ou dimensão. Foi considerado na análise um conjunto de I_{pq}^i , onde p representa o número de colunas, q o número de linhas, e i o número de imagens.

Já no trabalho de Lu, Plantaniotis e Venetsanopoulos [53], é proposto um novo método para LDA com o objetivo de diminuir o problema de pequenos conjuntos de dados de entrada para o treinamento, denominado *Small Sample Size* (SSS). Tal problema é apontado como o ponto fraco da LDA, pois devido a ele técnica não retorna bons resultados na classificação. Os autores tratam do contexto de reconhecimento de face, onde seu método, denominado *Regularized LDA* (R-LDA) para tentar otimizar o critério de regularização de Fisher.

Lu, Plantaniotis e Venetsanopoulos [53], discutem ainda, que para resolver o problema do SSS expandindo a capacidade representacional da base de dados disponível para o treinamento. Por exemplo, dado um par de protótipo que pertencem a mesma classe pode-se aplicar uma técnica denominada *Nearest Feature Line* (NFL – Linha Característica Mais Próxima) para virtualmente generalizar um número infinito de variações de dois protótipos sob variações de iluminação e de expressão. Além disso, em casos que não se tem o par de protótipos, pode-se aplicar uma técnica que rotaciona (gira) e traduz a imagem de protótipo para aumentar o número de amostras.

Li e Yuan [54] propõem um algoritmo para extração e seleção de imagem denominado 2D-LDA, executa a extração de características em uma matriz de imagens ao invés de utilizar vetores. O método proposto Análise Discriminante Linear de Fisher para aumentar o efeito da variação causada por diferentes indivíduos, com exceção da iluminação, expressão e orientação. O algoritmo calcula a matriz de dispersão entre classes e a matriz de dispersão dentro das classes. Em comparação com os algoritmos 2D-PCA, Eigenfaces e Fisherface o algoritmo proposto pelos autores apresentou uma maior precisão no processo de reconhecimento, além de apresentar um menor custo de computação comparado aos algoritmos testados.

Os trabalhos apresentados nos últimos parágrafos são exemplos que mostram como a LDA vêm sendo aplicada em problemas de classificação de objetos, e conseqüentemente, em técnicas de reconhecimento de padrões e reconstrução de imagens. Os trabalhos destacam ainda como alguns pontos fracos e como os mesmos têm sido contornados reforçando sua aplicação na classe de problemas de classificação de objetos.

4.2 Metodologia para Aplicar a Análise Discriminante

Conforme mencionado anteriormente, a análise discriminante é aplicada sob um dado domínio tendo como objetivo identificar e classificar membros (objetos) de grupos previamente estabelecidos [55]. Para aplicar a LDA é preciso seguir um conjunto de seis estágios, apresentados no diagrama de atividades da Figura 4.2. No diagrama, as atividades são discriminadas detalhadamente as quais inicia-se com a identificação das variáveis e observação dos grupos, e finaliza-se com a atividade de validação dos resultados discriminantes. Os estágios do diagrama foram detalhados da seção 4.2.1 até a seção 4.2.6.

4.2.1 Primeiro Estágio

No primeiro estágio é necessário analisar qual o objetivo de se aplicar a LDF ao problema desejado, pois a LDF é utilizada para atender um dos quatro objetivos: determinar se existem diferenças estatísticas significantes na pontuação média dos perfis sob um conjunto de variáveis para dois ou mais conjuntos pré-definidos, determinar quais das variáveis independentes contam mais para as diferenças de médias na pontuação de dois ou mais grupos, estabelecer o número e a composição de dimensões da discriminação de grupos formados a partir de um conjunto de variáveis independentes, e estabelecer procedimentos para a classificação de objetos em grupos baseando-se nas suas pontuações sob um conjunto de variáveis independentes.

A LDF deve executar ou uma análise de perfil, para entender as diferenças entre os grupos – baseado nas variáveis independentes –, ou uma técnica de previsão analítica, que executa uma classificação sob um conjunto de objetos, separando as observações em grupos previamente definidos. Portanto, esse estágio no procedimento de aplicação da LDF é necessário para identificar se o problema pode ser resolvido através da utilização da LDF e qual é o objetivo da aplicação do método.

4.2.2 Segundo Estágio

Neste estágio é necessário identificar os padrões na população (amostra) que se deseja analisar. Para isto, executa-se a identificação das variáveis dependentes e das variáveis independentes. Além disso, é necessário verificar o tamanho da amostra para estimar as funções discriminantes, bem como tais amostras serão divididas para a identificação do propósito de aplicação da LDF.

O primeiro passo nesse estágio envolve identificar as variáveis dependentes e as variáveis independentes. As dependentes são utilizadas para nomear os diferentes grupos que receberão os conjuntos de objetos. As independentes são responsáveis por nortear em qual grupo cada um dos objetos será classificado (irá pertencer), portanto, as variáveis independentes são chamadas variáveis métricas.

Cada um dos objetos deve pertencer a um e somente um grupo, ou seja, os grupos devem ser mutuamente exclusivos. Vale ressaltar que a LDF pode trabalhar com um número ilimitado

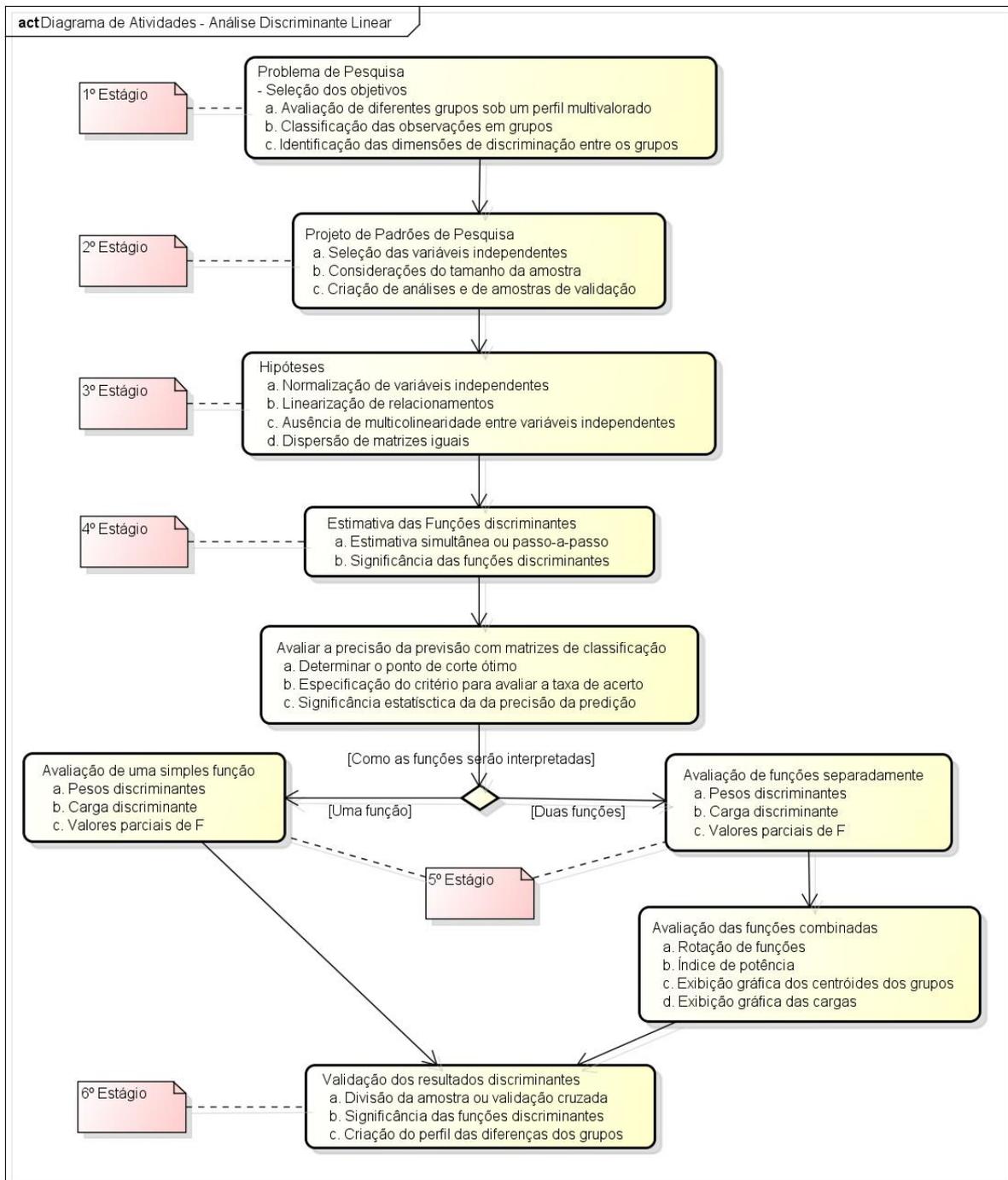


Figura 4.2: Processo contendo as atividades para aplicação da Análise Discriminante Linear (adaptado de [51]).

de variáveis dependentes, ou seja, com um número ilimitado de grupos. Desta forma pode-se selecionar o número de variáveis levando em consideração alguns aspectos, tais como, a garantia de que os grupos possam ser totalmente exclusivos, e a tentativa de se balancear o número total de grupos. Neste último caso, quanto mais grupos forem definidos, maior será a complexidade no processamento de tarefas de análise de perfis e classificação.

Após identificados os grupos e as métricas, para dividir e classificar os objetos da população em cada um dos grupos, deve-se passar para o segundo passo no estágio, que é a análise do tamanho das amostras. Esse procedimento é importante, pois, o tamanho das amostras interfere na qualidade dos resultados. Amostras pequenas não levam a identificação de grandes diferenças entre o conjunto de objetos analisados. Todavia, um número elevado de amostras, estatisticamente leva a identificação de diferenças significantes, aumentando a qualidade dos resultados. Pode-se levar em consideração duas análises quanto ao tamanho das amostras, o tamanho global das amostras e o tamanho das amostras por categoria (grupo a grupo).

No caso do tamanho global, conforme Hair e Anderson [51], a análise discriminante é totalmente sensível a razão do tamanho da amostra com número de variáveis previstas, por tal motivo, muitos estudos sugerem uma razão de 20 observações para cada variável prevista, ou uma razão de 5 amostras para cada variável independente. Um outro tipo de análise, o pesquisador pode considerar o tamanho das amostras por categoria, onde o tamanho do menor grupo de categorias deve exceder o número de variáveis previstas.

4.2.3 Terceiro estágio

Neste estágio são desenvolvidas as hipóteses da análise discriminante, onde devem ser considerados os tipos de hipótese e os impactos da aplicação da LDF. As hipóteses estão relacionadas com os processos estatísticos envolvidos nas estimativas e com os procedimentos de classificação utilizados na interpretação dos resultados.

No que diz respeito aos impactos com a estimativa deve-se atentar aos dados que não atendam a hipótese de normalidade multivariada, os quais podem trazer problemas no processo de estimativa da LDF. Neste caso, pode ser necessário aplicar soluções através de modificações nos dados para diminuir a disparidade nas matrizes de covariância. No caso do processo de classificação, matrizes de covariância desiguais podem afetar o resultado negativamente. Tamanhos pequenos de amostras e matrizes covariância desiguais causam problema de significância estatística, afetando o processo de classificação. Para diminuir os impactos negativos pode ser necessário aumentar o número de amostras analisadas.

4.2.4 Quarto Estágio

Este estágio envolve a derivação das funções discriminantes, onde deve ser decidido o método de estimativa e o número de funções utilizadas, e também o modelo de avaliação do ajuste global, que é responsável por identificar a precisão da classificação e seu impacto sobre o modelo de estimativa. Nesse estágio ocorre comparação dos centroides de cada grupo com as pontuações discriminantes Z (pontuação calculada para cada um dos objetos), de forma a fornecer uma medida de discriminação entre os grupos. Esse processo faz a medição da precisão do método de classificação.

Para derivar as funções discriminantes é preciso escolher o método de estimativa, que pode ser feito de duas formas, simultâneo ou passo a passo. No primeiro método todas as variáveis

independentes são calculadas concorrentemente no momento da execução do cálculo da função discriminante, ou seja, a função é calculada baseada em todo o conjunto de variáveis independentes. Este método enfatiza o poder de discriminação de cada uma das variáveis independentes no cálculo da função. Já o método passo a passo, as variáveis independentes são analisadas uma por vez com base em seu poder de discriminação, de forma que se possa escolher pela adição ou exclusão de uma variável da função discriminante.

Após o processo de estimativa das funções discriminantes, é necessário avaliar o nível de significância estatística do poder discriminatório de cada uma das funções (uma a uma e o conjunto como um todo). Quando o método de análise é executada sob um conjunto como o todo, a avaliação tem como objetivo testar a habilidade das funções discriminantes em derivar pontuações Z que são significativamente diferentes entre os grupos. Quando o número de grupos é superior a três é necessário analisar cada uma das funções discriminantes verificando se cada uma é estatisticamente significativa.

O número de funções discriminantes estimadas é obtido pelo número de grupos menos um, ou seja, no caso de N grupos, o número de funções NF será obtido através do cálculo $NF = N-1$. Se uma ou mais funções forem consideradas com baixa significância estatística será necessário re-estimar o modelo com o número de funções equivalente ao número de funções significantes.

O próximo passo neste estágio envolve a determinação do ajuste das funções mantidas para a análise discriminante. Este processo engloba o cálculo das pontuações discriminantes Z para cada uma das observações, simplificando o processo de interpretação e ainda avaliando a contribuição das variáveis independentes. Esse processo envolve ainda a avaliação das diferenças dos grupos baseado nas pontuações discriminante Z , uma vez que a pontuação distingue em qual grupo o objeto será alocado. Em uma última análise, o processo faz a avaliação dos membros do grupo quanto a precisão da classificação, sendo que as pontuações discriminantes podem ser utilizadas como um instrumento para prever os membros de cada um dos grupos.

Após aplicar a função discriminante para proceder com a classificação, pode-se verificar visualmente como os grupos foram formados, através de uma representação gráfica. Pode-se verificar quais elementos ficaram em grupos distintos e quais ficaram em um ponto de intersecção entre os grupos, ou seja, aqueles que ficaram sobrepostos – cuja suas características não foram suficientes para que o método de classificação conseguisse classificá-los em apenas um dos grupos [51]. A Figura 4.3 ilustra a distribuição das classe em um plano.

A Figura 4.3, apresenta a análise de dois grupos através do cálculo da função discriminante, em que A é um grupo representado por quadrados e B é um grupo representado por elipses. V_1 e V_2 , representam a medida de cada membro de ambos os grupos no plano. Z é um novo eixo formado pela medida da função discriminante que expressa dois perfis variáveis com os números das medidas. A' e B' representam as medidas obtidas pela função discriminante, e a parte sombreada é a sobreposição dos mesmos.

4.2.5 Quinto Estágio

Este estágio é responsável pela interpretação dos resultados, verificando se a LDF é estatisticamente significativa e se a precisão da classificação é aceitável. Este processo envolve avaliar as funções discriminantes para determinar a importância de cada variável independente no processo de discriminação entre os grupos [51]. São aplicados três métodos para avaliar a im-

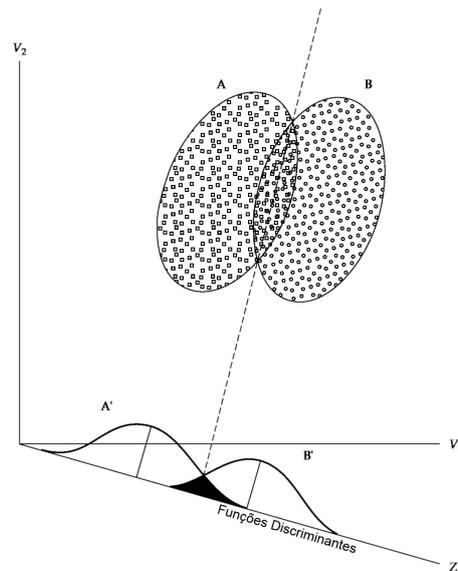


Figura 4.3: Exemplo da análise de dois grupos através do cálculo da função discriminante, em que A é um grupo representado por quadrados e B é um grupo representado por elipses. V_1 e V_2 , representam a medida de cada membro de ambos os grupos no plano. Z é um novo eixo formado pela medida da função discriminante que expressa dois perfis variáveis com os números das medidas. A' e B' representam as medidas obtidas pela função discriminante e a parte sombreada é a sobreposição dos mesmos [51].

portância das variáveis, os pesos discriminantes padronizados, cargas discriminantes (estrutura de correlações), e os valores F parciais.

O coeficiente discriminante (ou peso discriminante) é associado a cada uma das variáveis utilizadas no cálculo da função discriminante. Cada um desses pesos tem o seu sinal e sua magnitude avaliados, sendo que na análise quando o sinal é ignorado cada peso representa a contribuição relativa da variável para a função, ou seja, maiores pesos levam a maiores contribuições no poder de discriminação da função. O sinal denota se a contribuição da variável é positiva ou é negativa. Um grande problema desse método de análise é o fato de que a definição dos pesos é feito de forma subjetiva podendo causar problemas na interpretação dos resultados.

A carga discriminante vem sendo mais utilizada por causa da deficiência na definição e utilização dos pesos. Nesse método é medido a correlação linear entre as variáveis independentes e a função discriminante, sendo que as cargas discriminantes refletem a variância que as variáveis independentes compartilham com a função discriminante. Essa medida avalia a contribuição relativa de cada variável independente com a função de discriminação. Esta medida pode ser calculada para todas as variáveis independente de sua utilização ou não na função discriminante. Este método é considerado mais válido do que a análise de pesos na interpretação do poder de discriminação de variáveis independentes.

Duas abordagens computacionais podem ser utilizadas para derivar funções discriminantes, a abordagem passo a passo e a abordagem simultânea. No passo a passo é possível usar o método de denominado valores parciais de F para interpretar o poder de discriminação relativo das variáveis independentes. Esse método avalia o tamanho absoluto dos valores significantes

de cada função F e faz um ranking (posicionamento) do poder discriminatório de cada uma das funções F .

4.2.6 Sexto Estágio

Esse é o último estágio do processo de análise discriminante, onde os resultados são validados de forma a garantir a validade interna (sob as amostras analisadas) e externa (sob um conjunto generalizado de amostras). Uma das formas de validação externa é através da avaliação dos índices de sucesso. A validação pode utilizar um conjunto de amostras de validação, que é a forma mais comum de validação e trabalha com observações que não foram utilizadas para derivar a função discriminante, ou pode utilizar um procedimento que processa repetidamente as amostras estimadas, que utiliza diversos subconjuntos da amostra total para validar os resultados e têm as funções discriminantes calculadas para cada subconjunto de amostras.

4.3 Utilização do LDF

Com base na metodologia apresentada, sua aplicação na resolução de problemas de classificação supervisionada pode ser realizada em dois passos: treinamento e classificação. Ambos os passos são apresentados detalhadamente a seguir.

4.3.1 Processo de Treinamento

O processo de treinamento é baseado fortemente na obtenção da matriz de características dos elementos, que formam as classes e que serão utilizados no treinamento e nos vetores de pesos ótimo. A seguir são apresentadas as equações necessárias para a obtenção da matriz de características e dos vetores de pesos ótimo [56]. Para um conjunto K de classes relacionadas ao problema de reconhecimento de padrões, cuja as características são organizadas em vetores com dimensão M . Sendo denotadas pela Equação 4.2:

$$X_i = [x_{i,1} x_{i,2} \cdots x_{i,N_i}]^T \quad (4.2)$$

a sequência de vetores de características correspondem a uma determinada classe i , onde X_i é uma matriz $N_i \times M$, e N_i é um número de vetores de características que pertencem a classe i . Concatenando as matrizes X_i para todo o conjunto K de classes resulta-se na Equação 4.3:

$$X = [X_1 X_2 \cdots X_K]^T \quad (4.3)$$

Considerando que y_i seja um vetor de saída ideal e pertencente a classe i , que é uma coluna do vetor constituída de zeros e uns, tal que: $y_i = [0_{N_1}, 0_{N_2}, \cdots, 0_{N_{i-1}}, 1_{N_i}, 0_{N_{i+1}}, \cdots, 0_{N_K}]$. O processo de treinamento consistirá apenas no cálculo do vetor de peso ótimo denotado por w_i^{opt} que reduz a distância entre y_i e uma combinação linear dos vetores de características treinado Xw_i , tal que:

$$w_i^{opt} = \arg_{w_i} \min \| Xw_i - y_i \|_p \quad (4.4)$$

A Equação 4.4 indica que o vetor de peso ótimo que poderia ser obtido por minimização da norma L^p do vetor de erro $e_i = Xw_i - y_i$. A minimização da função apresentada na Equação

4.3 utilizando a norma L^2 conduz a Equação 4.5:

$$w_i^{opt} = (X^T X)^{-1} X^T y_i \quad (4.5)$$

Logo, na Equação 4.5 é apresentada uma equação simplificada para ser utilizada no treinamento, isto com base nas características intrínsecas dos objetos das diversas classes.

4.3.2 Classificador LDF

Após o treinamento a utilização do classificador LDF é simples. Considerando X_C como um vetor de características que necessita ser classificado em um dos conjuntos de classes K , o processo pode ser realizado com base na saída da avaliação do vetor de características comparado com todos os modelos de K , calculando uma série de *scores* s_i , tais como apresentado na Equação 4.6:

$$s_i^{opt} = X_C w_i^{opt} \quad (4.6)$$

Logo, a classe da sequência X_C é determinada pela seleção da classe com a maior pontuação, assim com apresentado na Equação 4.7:

$$c = arg_{i} max(s_i) \quad (4.7)$$

Capítulo 5

Sistemas de Rastreamento Baseado em PSO e LDF

Este capítulo apresenta a proposta de um sistema de rastreamento visual otimizado para aplicações com baixo custo computacional. Para este propósito, faz uso das estratégias do método de PSO em seu rastreamento, além de contar com a função de custo sendo realizada pela técnica LDF, que através de histograma RGB (*Red, Green, Blue*) de blocos (conjunto de *pixels*) definem as classes objeto alvo e não alvo do rastreamento. Inicialmente é apresentado o algoritmo da proposta, em seguida é apresentada a divisão dos quadros para inicialização do sistema, a obtenção do fluxo óptico, para servir de entrada para o método de PSO, a metodologia do treinamento do LDF, para se utilizar como função de custo e, por fim, detalhar as contribuições do trabalho proposto, que é a aplicação do LDF baseado em histogramas RGB para proporcionar a função de custos para o rastreador visual implementado com o método de PSO.

Em trabalhos anteriores [1], [2], [3] [4], a utilização do PSO para o rastreamento visual se demonstrou eficiente, chamando a atenção para este método de otimização. Devido as características de rápida convergência, baixo custo computacional, capacidade de tratar pequenas oclusões, movimentos arbitrários e movimentos abruptos.

A medida que foram aprofundados os conhecimentos na técnica do PSO foi identificada que a precisão do algoritmo se dava pela função de custo adotada pelo sistema. John, Trucco e Ivekovic [4] apresenta um algoritmo hierárquico que utiliza o PSO para rastreamento baseado em movimentos do corpo humano, com função de custo baseada na função Somatória das Diferenças Quadráticas (*Sum Squared Differences - SSD*).

Já Hsu e Dai [1] implementou o rastreamento baseado no PSO para múltiplos alvos também buscando a utilização em aplicação de tempo real, em que foi utilizada a diferença entre histogramas em escalas de cinza para comparar cada partícula no espaço de busca com o objeto alvo e múltiplos enxames. Apesar do método criado demonstrar boa eficiência foi apresentada a necessidade de maior precisão nas buscas por características dos alvos que estão sendo rastreados.

Zheng e Meng [3] também utiliza o método de PSO para realizar o rastreamento, mas a função de custo é baseada em histogramas acumulativos, porém antes de se obter o histogramas é feita a transformação do espaço de cores RGB para HSV (*Hue, Saturation, Value*), a técnica de mudança dos espaços de cores foi objetivando separar componentes estáveis de ruidosos.

Buscando o aumento da precisão do método de PSO é que foi adotado neste trabalho um mecanismo de classificação estatístico de baixo consumo de recursos computacionais, tal método

deveria possuir aprendizado supervisionado para aumentar o poder de classificação.

Logo, foram aplicados esforços no desenvolvimento, em Matlab, do sistema de rastreamento baseado no PSO e com função de custo do PSO baseada no LDF treinado, utilizando histogramas RGB para identificar as classes, alvo ou não alvo do rastreamento. A escolha pelo histogramas RGB, ocorreu devido as característica das cores em vídeos serem marcantes, além de possibilitar maior conjunto de dados (característica) para diferenciação das classes, alvo e não alvo, durante o processo de treinamento, uma vez que o desempenho da técnica LDF está ligada diretamente a quantidade do conjunto de dados [52].

A próxima seção aborda em detalhes o algoritmo do sistema proposto através de um fluxograma.

5.1 Algoritmo Proposto

A Figura 5.1 apresenta o algoritmo do sistema proposto através de um fluxograma.

O algoritmo do sistema proposto apresentado na Figura 5.1 é composto por uma fonte de entrada, que são quadros de um vídeo ou quadros obtidos em tempo real através de um equipamento de captura, por exemplo, filmadora posicionada em uma base fixa, câmera de monitoramento ou *webcam*.

O sistema inicia com uma variável booleana, chamada “*Pronto para Rastrear*”, que servirá de *flag* para a preparação do ambiente e do momento de rastreamento do visual, configurada com o valor falso. Pois, antes de iniciar o rastreamento se faz necessário obter o alvo e treinar a função de custo do rastreador desenvolvido. O sistema iniciará após o segundo quadro capturado, uma vez que para se obter o fluxo óptico que segmentará o objeto alvo são necessários pelo menos dois quadros.

No sistema proposto após a “*Captura do Quadro*” terá um condicional que avaliará a variável booleana, apresentada anteriormente, chamada “*Pronto para Rastrear*”.

Caso a variável “*Pronto para Rastrear*” esteja configurada com o valor falso, o sistema passará para o processo de comparação do quadro atual com o quadro anterior para se estimar movimento. Após esta comparação *pixel a pixel* é feito o processo de obtenção do fluxo óptico e, através deste fluxo óptico, é feita segmentação do objeto alvo para que o mesmo seja rastreado. Esta segmentação é mantida em forma de máscara, com a região do quadro destacada através de contornos. Caso não seja alcançado o alvo neste quadro, o sistema retorna para captura do vídeo para que se possa analisar um novo quadro do vídeo. Caso contrário, uma vez segmentado o quadro, têm-se a localização do alvo e do não alvo, permitindo, assim, o treinamento da função de custo do PSO. Através desta segmentação são distribuídas as partículas do PSO no quadro original sobre o alvo delimitado pela máscara, em seguida são distribuídas o dobro das partículas do PSO no quadro original, para treinar a função de custo do que não é alvo. Em seguida é realizado o treinamento da Função Discriminante Linear que em cada partícula alvo ou não alvo obterá um histograma RGB do bloco que servirá como características de entrada de cada grupo. Uma vez treinado o sistema de rastreamento, ele deverá ser capaz de classificar as novas medidas de custo em um dos dois grupos, alvo ou não alvo. Portanto, é necessário que a variável “*Pronto para Rastrear*” seja configurada com o valor verdadeiro e o sistema retorne para captura de novos quadros.

Caso esta variável esteja configurada com o valor verdadeiro, o sistema passará para o processo de rastreamento visual com PSO. Em sua primeira iteração no quadro possui as velo-

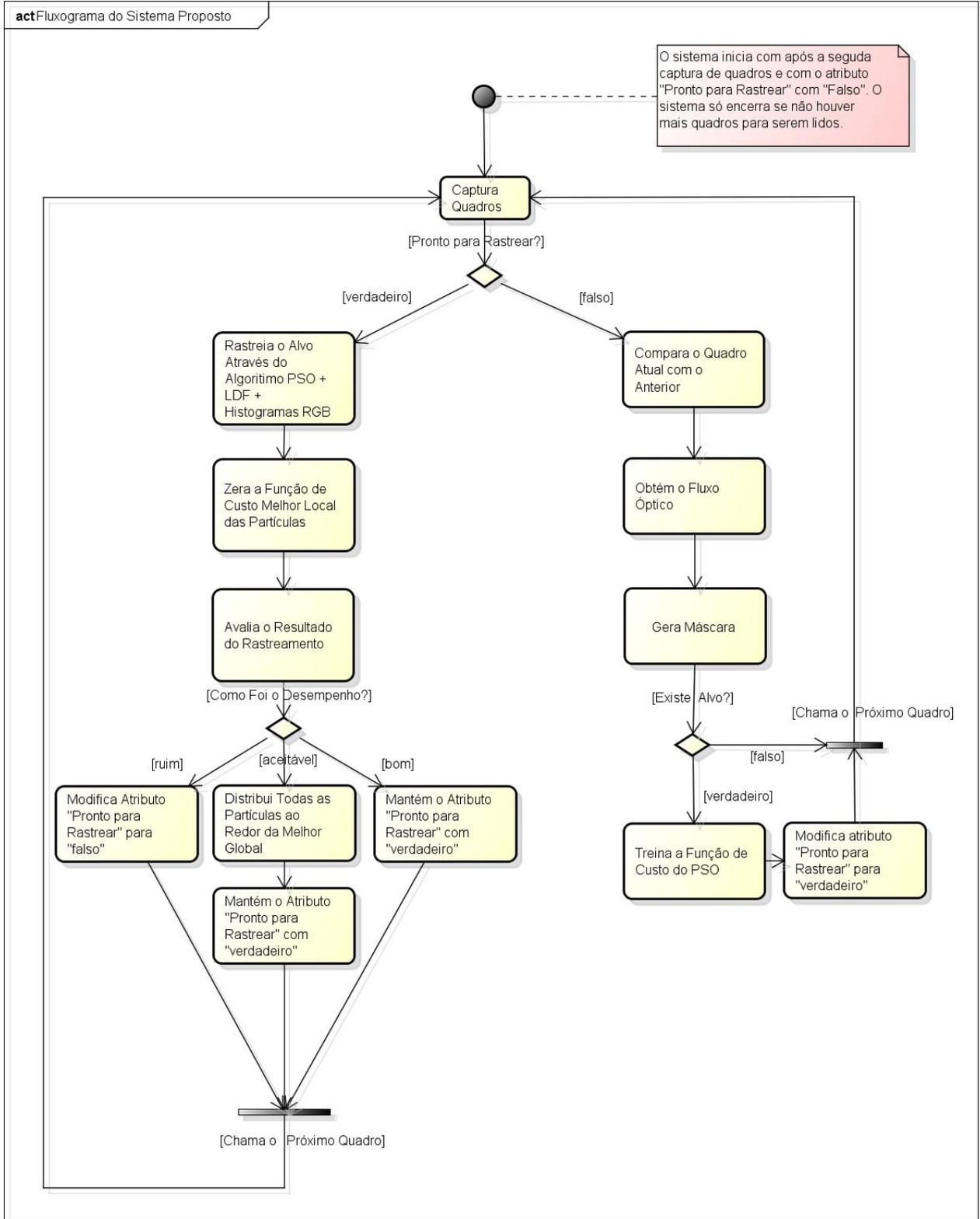


Figura 5.1: Fluxograma do Sistema Proposto.

idades aleatórias baseada na posição obtida durante o treinamento ou herdadas as velocidades do rastreador visual da última iteração do quadro anterior, caso o processo de rastreamento já tenha iniciado com outro quadro anterior. As posições das partículas do PSO são as mesmas obtidas no processo de treinamento ou são as das partículas da última iteração do quadro anterior, também, caso o processo de rastreamento já tenha iniciado com outro quadro anterior. Logo, com base nestas informações é calculada a posição do alvo na primeira iteração do quadro. Nas demais iterações do quadro será necessário calcular a função de custo de cada partícula, isto, através classificação do LDF que usa histograma RGB dos blocos. Isto, para que seja possível eleger a partícula com melhor posição global e a com a melhor posição local, que permitirá calcular a velocidade e realizar a atualização da posição das partículas em direção ao alvo, permitindo ao método convergir. Ao final de cada quadro, são realizadas duas ações: a limpeza da função de custo da melhor posição local e a avaliação do desempenho da função de custo da melhor posição global, que deverá estar sobre o alvo. Caso o valor avaliado esteja em um limite aceitável, apesar de poder estar próximo do insatisfatório, é realizada uma estratégia de recuperação do rastreamento. Tal estratégia realiza a redistribuição das partículas ao redor partícula com a melhor posição global. Após o uso desta estratégia a variável “*Pronto para Rastrear*” será configurada com o valor verdadeiro. Caso o valor avaliado seja bom, a variável “*Pronto para Rastrear*” será configurada com o valor verdadeiro. Caso o valor avaliado seja inaceitável a variável “*Pronto para Rastrear*” será configurada com o valor falso, obrigando o sistema a fazer uma nova obtenção do alvo e realizar um novo treinamento. Em seguida, o sistema retorna para captura de novos quadros.

Enquanto houver quadros de entrada para o sistema ler, o sistema ficará realizando o rastreamento visual.

A próxima seção apresenta em detalhes como funciona a divisão dos quadros que alimentam o sistema, tais quadros podem servir para obtenção do alvo/treinamento ou para o rastreamento visual.

5.2 Divisão dos Quadros de Treinamento com os de Rastreamento

Todas as tarefas do sistema proposto são baseadas essencialmente nos quadros de entrada, seja de um vídeo gravado ou obtidos em tempo real através de qualquer equipamento de captura. Inicialmente, os quadros seguem um dentre dois caminhos, ou ele será utilizado no rastreamento visual através do método de PSO, ou ele será utilizado na estimação de movimentos para obtenção do alvo e treinamento da função de custo. A Figura 5.2 apresenta o esquema de utilização dos quadros que vão alimentar o sistema proposto.

Como os processos de estimação de movimento e obtenção do fluxo óptico apresentam alto custo computacional, o objetivo do sistema proposto é o de reduzir ao máximo estas tarefas utilizando apenas quando for essencial, ou seja, quando o algoritmo de rastreamento baseado no PSO não conseguir mais rastrear o objeto alvo ou no início para a identificação do objeto alvo.

Logo, assim como apresentado na Figura 5.2 o conjunto de M quadros, que serão utilizados para a obtenção do objeto alvo e treinamento do rastreamento, caso haja um alvo para ser rastreado desde o início da captura dos quadros serão necessários apenas dois quadros nesta etapa, caso contrário o sistema ficará a espera de um alvo lendo M quadros capturados até que

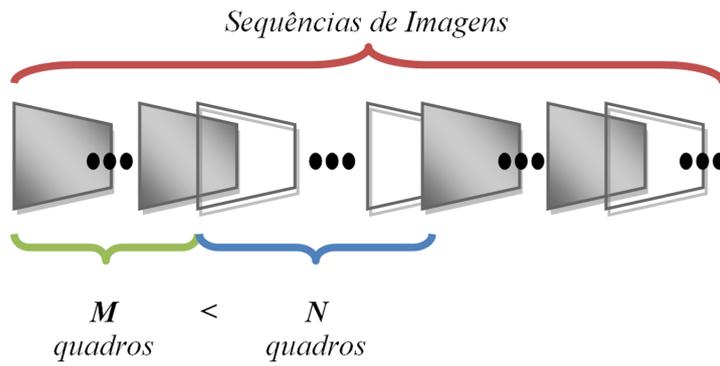


Figura 5.2: Esquema de utilização dos quadros de entrada do sistema proposto, onde uma sequência de imagens são divididas em M quadros e N quadros, onde $M < N$ e M servirá de entrada para o processo de estimação de movimentos e N servirá de entrada para o processo de rastreamento.

surja um objeto alvo. Um vez que exista um objeto alvo para ser rastreado os N quadros serão utilizados nesta tarefa. Após um período, caso o rastreado não consiga mais rastrear o objeto alvo ou após um determinado tempo escolhido, o conjunto de quadros M permitirá reencontrar o alvo e realizar um novo treinamento. Em um ambiente que constantemente existe um alvo, espera-se que o conjunto de quadros M seja bem inferior ao conjunto de quadros N , garantindo, assim, a redução de custo computacional de forma elevada, para o propósito de rastreamento.

Na próxima seção são abordados alguns conceitos utilizados para obtenção do objeto alvo, que posteriormente será rastreado. Tais conceitos envolvem estimação de movimento e fluxo óptico.

5.3 Estimação de Movimento e Fluxo Óptico

A segmentação de movimentos em vídeo digital está basicamente relacionada com a extração de movimentos em vídeos, em que, além de permitir a identificação de movimentos, permite a modelagem de mudanças ocorridas em regiões da imagem [57].

O uso do movimento na segmentação consiste na análise de dois quadros obtida no momento t_i e t_j ($j < i$). Assim como é apresentado na Figura 5.3, porém identificados como quadro de referência e quadro atual, respectivamente.

Com base na Figura 5.3 uma determinada região no quadro atual, conhecida como macrobloco atual é selecionada. Este bloco é comparado (*pixel a pixel*) com blocos do mesmo tamanho no quadro de referência dentro de uma janela de busca. O bloco mais similar no quadro de referência é conhecido como predito e a diferença entre as posições do bloco predito e original é o vetor de movimento. Se o vetor de movimento é associado ao bloco original inteiro então pode-se dizer que a estimação de movimento foi feita bloco a bloco, por outro lado se o vetor de movimento é associado somente ao *pixel* central do bloco então a estimação de movimento está sendo realizada no nível de *pixels* [6].

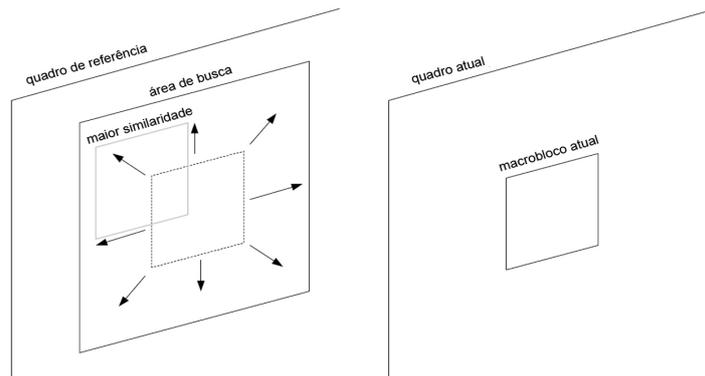


Figura 5.3: Ilustração do processo de estimativa de movimentos, onde são apresentados dois quadros um de referência e outro que é o quadro atual, que terá seu macro bloco comparado com uma área de busca do quadro de referência e com base no local com de maior similaridade é registrada a diferença entre eles [6].

Geralmente, os movimentos de um quadro a outro são muito pequenos em um sinal de vídeo que possui entre 25 a 30 quadros por segundo, logo a análise deve ser sensível para capturar os movimentos, porém resistente o bastante para ignorar os ruídos.

Fluxo Óptico, no entanto, é a distribuição bidimensional da velocidade aparente do movimento dos padrões de intensidade, relacionados a objetos, superfície e regiões, no plano da imagem [58]. Geralmente, o resultado do campo de visão do fluxo óptico é apresentado em uma malha com traços que representam a velocidade, chamado de “diagrama de agulhas” ou *needle map* na língua inglesa. Um fluxo óptico completo pode ser visto como os vetores de movimentos obtidos de uma estimativa de movimento *pixel a pixel*.

Segundo trabalhos anteriores [59], o fluxo óptico pode ser importante para a realização das seguintes atividades: interpretação de cena, navegação exploratória, acompanhamento de objetos, avaliação de tempo para colisão, segmentação de objetos, codificação de vídeo, visão de robôs, entre outras. Neste trabalho, o fluxo óptico obtido mediante uma estimativa de movimento será utilizado para segmentar as regiões com mais movimento dentro de uma sinal de vídeo. Já que numa cena de um ambiente monitorado, as regiões de maior movimento representam os objetos de maior interesse.

A Figura 5.6 apresenta a diferença entre os quadros apresentados na Figura 5.4 e na Figura 5.5, em que a região cinza representa a diferença sendo zero, a região mais clara representa a diferença sendo positiva e a região mais escura representa a diferença sendo negativa. Já a Figura 5.7 apresenta o fluxo óptico por blocos obtido dos mesmos quadros. Pode-se observar que as regiões com maior magnitude dos vetores de movimento, são justamente os lugares onde a diferença entre os quadros também são maiores.

Na próxima seção são apresentados os detalhes para obtenção do objeto alvo, que posteriormente será rastreado. Para obtenção do objeto alvo é realizada a estimativa de movimentos, que oferecerão os vetores de movimento utilizados pelo fluxo óptico para se obter a segmentação da região do objeto alvo.



Figura 5.4: Quadro 1 do exemplo para demonstração do processo de estimação de movimento [6].



Figura 5.5: Quadro 2 do exemplo para demonstração do processo de estimação de movimento [6].

5.4 Detecção do Objeto Alvo

A detecção do objeto alvo através de estimação de movimentos é um processo muito comum na área de visão computacional. Há diversos algoritmos que podem ser utilizados para a tarefa de se obter a estimação de movimento e o fluxo óptico, no entanto foi utilizado um algoritmo muito comum que é o de Horn-Schunck [58]. Durante o processo foi identificado que quadros que apresentavam muitos ruídos, na máscara gerada a partir do fluxo óptico, deveriam receber um tratamento para otimizar a identificação do alvo. O tratamento consistia no processo de realizar erosão e dilatação da máscara obtida.

A Figura 5.8 apresenta a máscara baseada em quadros aleatórios do Video1.avi.

Na próxima seção é apresentada a metodologia adotada no treinamento da função de custo que permitirá ao rastreador baseado no PSO maior precisão em seu funcionamento.

5.5 Treinamento da Função de Custo

Ao optar por uma aplicação baseada em classificação supervisionada, o treinamento é uma fase muito sensível e importante no processo, pois ele é que irá fornecer a capacidade do sistema em identificar a qual grupo pertence determinado elemento.

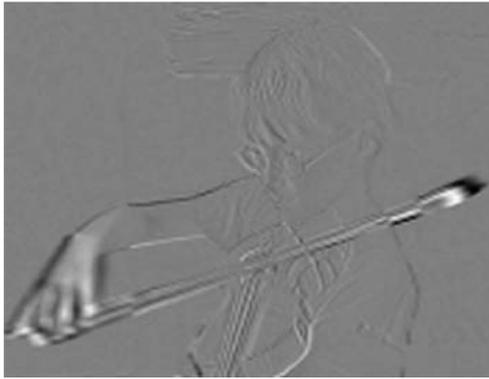


Figura 5.6: Resultado da diferença dos quadros 1 e 2 do exemplo para demonstração do processo de estimação de movimento [6].



Figura 5.7: Fluxo Óptico obtido com base na diferença dos quadros 1 e 2 [6].

No sistema proposto foi adotada a técnica LDF, que possui como entrada diversas características dos grupos que serão classificados. No sistema proposto haverá apenas duas classes, a alvo e a não alvo e os elementos que servirão de entrada serão escolhidos aleatoriamente com base em partículas (*pixels*) dentro na área segmentada na máscara, porém aplicadas no quadro, e com partículas (*pixels*) fora da área segmentada na máscara, porém aplicadas no quadro, também.

As características escolhidas para representar as classes alvo e não alvo foram histogramas RGB de blocos de tamanhos variados (8×8 *pixels* e 16×16 *pixels*) ao redor de cada partícula, que possuíam 256 valores para cada canal do histograma, gerando como entrada 768 características para cada partícula representando um elemento de uma das classes. O tamanho do conjunto de partículas para o treinamento foi variado, porém é baseado nos valores apontados antes de iniciar o sistema de rastreamento baseado no PSO que será identificado como $n \times 3$. Logo, a quantidade de partículas para identificar o grupo de alvos foi n e a quantidade de partículas para identificar o grupo de não alvos foi de $n \times 2$.

Antes de iniciar o treinamento o sistema obtém um histograma de cada canal do bloco ao redor da partícula, tanto da classe alvo, como da classe não alvo, formando, assim, o vetor de características de cada partícula, sendo acrescentado apenas os valores '1' no início do vetor de características e a média de todas as características ao final do vetor de características. Após esta atividade em todas as partículas, é obtida a matriz de características de todos os elementos

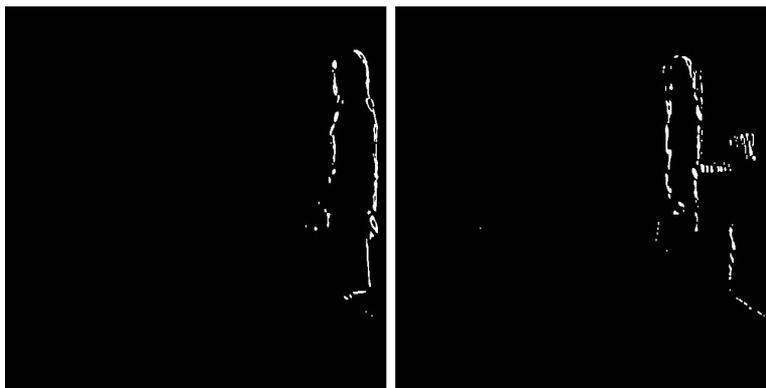


Figura 5.8: Exemplo de máscara baseada em quadros aleatórios do Video1.avi.

do treinamento. Além desta matriz é necessário obter o vetor de saída ideal, que para todas as n partículas da classe alvo foi inserido '0' e para todas as $n \times 2$ partículas da classe não alvo foi inserido o valor '1'. A Figura 5.9 apresenta um exemplo do vetor de características de entrada que exemplifica as entradas alvo e não alvo.

Índice	1	2	3	4	5	6	...	765	766	767	768	769	770
Alvo	1	28	0	0	0	0	...	1	1	1	1	1	1
Não Alvo	1	1	0	0	0	0	...	1	1	2	5	237	1

Figura 5.9: Exemplo do vetor de características de entradas, classe alvo e não alvo, onde a primeira linha apresenta o vetor de características do objeto alvo e a segunda linha apresenta o vetor de características da região não alvo, extraídas de histogramas de cada canal a partir da coluna 2, indo até a coluna 769, a coluna um apresenta um valor padrão 1 para todas as linha e a coluna 770 possui a média das posições anteriores.

A Figura 5.10 apresenta o histograma de todos os canais das entradas alvo e não alvo do treinamento da função de custo.

Com base, nestas duas entradas, a matriz de características e o vetor de saída ideal de todos os elementos do treinamento, utilizando a Equação 4.5, são obtidos os cálculos do vetor de peso ótimo, que, assim como apresentado anteriormente, minimiza a distância entre o vetor de saída ideal e a combinação linear dos vetores de características. Logo, como resultado do processo, haverá dois vetores de pesos com 770 valores que permitirão a classificação, de alvo ou não alvo, de uma nova partícula apresentada ao rastreador.

Na próxima seção é apresentado o rastreador baseado no PSO, com a classificação da função de custo baseado no LDF treinado que usa como características histogramas RGB baseados em blocos.

5.6 PSO + LDF + Histograma RGB

O rastreamento utilizando o método de PSO não é novidade, outros trabalhos abordam a utilização do PSO, assim como apresentado anteriormente. Porém, a principal contribuição desta pesquisa é a utilização do método de PSO com funções de custo baseada em uma técnica de

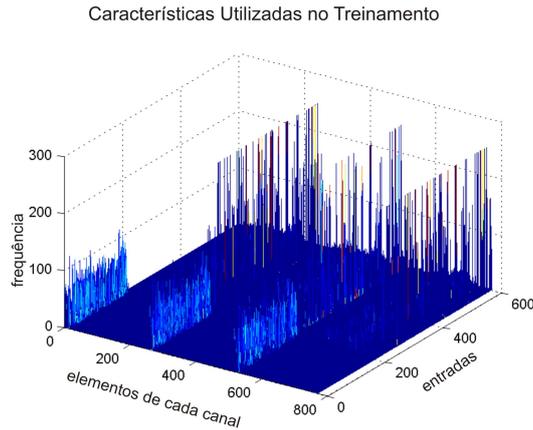


Figura 5.10: Exemplo do vetor de características de entradas, classe alvo e não alvo, onde no eixo x (abscissa) é apresentada a frequência que é o valor que será armazenado em cada posição do vetor identificado pelo eixo y (afastamento), que vai representar cada elemento de cada canal de cor e no eixo z (cota) é representada cada entrada da classe alvo, de 0 a 200, e da classe não alvo, de 200 a 600, neste exemplo.

classificação supervisionada, LDF, utilizando histogramas RGB de blocos do quadro para caracterizar o alvo. A seguir é apresentada a visão geral do algoritmo de rastreamento proposto.

Após o treinamento, o rastreador proposto (PSO + LDF + Histograma RGB) necessitará de um novo quadro para executar a sua primeira iteração, todas as posições das partículas que estavam sobre o alvo e a função de custo treinada. Em seguida é obtida a função de custo de todas as partículas do PSO no quadro atual, neste processo são necessárias 770 características de entrada, obtidas através de histogramas RGB do bloco onde se encontra a partícula e a média de todas as características, este vetor de características da partícula deve ser multiplicado com o vetor de peso ótimo da classe alvo, obtendo, assim, um valor. Na função de custo quanto mais próximo o valor estiver de '1' mais próximo ele estará do alvo.

Em posse da função de custo de todas as partículas do enxame, é necessária a comparação entre elas, a partícula que possuir o melhor valor na função custo, baseada na aproximação ao valor '1', será eleita a melhor posição global, necessitando informar para todas as partículas a sua função de custo e sua posição atual. Neste primeiro momento a melhor posição local de cada partícula será obtida pela posição anterior da partícula, que é gerada pela distribuição aleatória da partícula no momento do treinamento.

A próxima posição da partícula x_{k+1} é obtida com base na posição atual da partícula calculada com a velocidade aleatória, tal atualização da posição é baseada na Equação 3.8. Ao final da iteração a velocidade é atualizada com base na melhor posição local, melhor posição global e na velocidade anterior, tal atualização é baseada na Equação 3.9.

No início de cada nova iteração é necessário o tratamento de borda, que reposicionam as partículas horizontalmente e/ou verticalmente dentro do limite, para evitar que as partículas extrapolem os limites do quadro ou não consigam obter o tamanho de bloco adequado para calcular a função de custo. Posteriormente, deve-se calcular a função de custo de todas as partículas, utilizando o mesmo método apresentado anteriormente.

A melhor posição local de cada partícula será obtida pela comparação do valor da função de custo da partícula atual com o valor da função de custo da melhor posição local na iteração

anterior multiplicado com um valor X . X é maior do que 0 e menor do que 1 e assume a tarefa possibilitar a atualização da melhor posição local quando uma partícula atual possui o mesmo valor de custo da melhor posição local na iteração anterior. Após a comparação, se o valor da função de custo da posição atual for superior a melhor posição local, a melhor posição local receberá os valores desta partícula, posição e valor da função de custo.

Em seguida é verificado o maior valor obtido na função de custo dentre todas as partículas da iteração, a posição e o valor da função de custo da partícula que obteve maior função de custo na iteração são copiados para melhor posição global. A próxima posição da partícula x_{k+1} é calculada com base na Equação 3.8, utilizando a posição atual e a velocidade calculada na iteração anterior. Após realizar a atualização da posição de todas as partículas deve-se atualizar a velocidade. A velocidade é calculada utilizando a melhor posição local, melhor posição global e na velocidade anterior, tal atualização é baseada na Equação 3.9.

Ao final da última iteração no quadro, deve-se realizar a “limpeza” no valor da função de custo da melhor posição local de todas as partículas, atualizando-a com o valor ‘0’, tal “limpeza” busca garantir a atualização da mesma na primeira iteração no novo quadro, que possivelmente teve o alvo deslocado.

Em seguida é necessária a avaliação do desempenho do valor da função de custo da melhor posição global, que deverá estar sobre o alvo. Caso o valor avaliado esteja em um limite aceitável, será realizada uma estratégia de recuperação do rastreamento do alvo, evitando que o enxame de partículas perca o alvo, e, em seguida, é feita a leitura do próximo quadro. A estratégia de recuperação do rastreamento do alvo consiste na redistribuição das partículas ao redor partícula com a melhor posição global. Caso a avaliação seja boa, perante o patamar estabelecido é lido um novo quadro reiniciando todo o processo. Caso contrário é iniciado um novo processo de detecção do alvo e um novo treinamento.

No próximo capítulo são apresentados os resultados obtido com o algoritmo de rastreamento proposto, onde serão demonstradas as configurações utilizadas nos testes, os resultados dos rastreamentos e os teste de desempenho e complexidade, relacionado a métodos tradicionais.

Capítulo 6

Resultados

Neste capítulo são apresentados detalhes dos testes realizados no sistema de rastreamento proposto. Inicialmente, são identificados e apresentados todos os parâmetros utilizados nos testes de rastreamento, em seguida, são abordados os resultados obtidos com a execução dos testes, são avaliados o desempenho e a complexidade do algoritmo proposto em relação a técnica tradicional de estimação de movimento por correspondência de janela. É abordado o teste com algoritmo proposto de forma completa e, por fim, são realizados testes do algoritmo proposto com adaptações para tratar oclusões.

O ambiente de testes contou com a utilização de computadores pessoais baseados na arquitetura de processamento Intel, com os modelos Core i5 ¹ e Core 2 Duo ², tendo cerca de 4 Ghz de memória de acesso aleatório (RAM). Os vídeos foram capturados através de uma câmera de uso doméstico que possui a resolução equivalente a 640×480 pixels.

Assim como apresentado no capítulo anterior, foi desenvolvido um algoritmo que implementa o método de PSO para executar o rastreamento visual dos objetos alvos (os indivíduos em movimento), juntamente com funções de custo para classificar com precisão de tal objeto, utilizando o classificador LDF com características baseadas em blocos de histogramas RGB.

Foi utilizado um conjunto de testes para analisar a eficiência do algoritmo proposto na execução dos processos de rastreamento de indivíduos em movimento e de identificação dos parâmetro que melhores se adequam ao método de PSO. Também foram utilizadas duas sequências de imagens com oclusões sobre o alvo, tendo como principal objetivo testar o algoritmo em situações adversas.

A Tabela 6.1 apresenta as principais propriedades dos vídeos, dentre elas, o nome do vídeo, a quantidade total de quadros, o tempo de duração em segundos e a quantidade de quadros por segundo (FPS - *Frames Per Seconds*). A lista abaixo apresenta as características do conteúdo de cada um dos vídeos quanto as cores, o fundo (*background*) e a distância do equipamento de captura para o alvo:

- O Video1.avi apresenta uma pessoa em movimento em uma sala, próxima ao equipamento de captura, onde a pessoa realiza movimentos abruptos com o corpo inteiro. Ao pular, e com um membro do corpo, ao mexer o braço rapidamente por um intervalo de tempo relativamente alto, as cores das vestimentas são parcialmente diferenciadas do fundo. A Figura 6.1 (a) apresenta um quadro que tipifica o Video1.avi.

¹<http://www.intel.com.br/content/www/br/pt/processors/core/core-i5-processor.html>

²http://www.intel.com/p/pt_BR/support/highlights/processors/core2duo

- O Video2.avi apresenta uma pessoa em movimento em uma sala, onde a pessoa realiza movimentos comuns de deslocamento em um único sentido. Nesse vídeo as cores das vestimentas são parcialmente diferenciadas do fundo. A Figura 6.1 (b) apresenta um quadro que tipifica o Video2.avi.
- O Video3.avi apresenta uma pessoa em movimento em um ambiente aberto de uma edificação, onde a pessoa realiza movimentos comuns de deslocamento em um único sentido. As cores das vestimentas são diferenciadas do fundo, nos pontos com o contato direto. A Figura 6.1 (c) apresenta um quadro que tipifica o Video3.avi.
- O Video4.avi apresenta uma pessoa em movimento em um ambiente aberto de uma edificação, nesse momento a pessoa realiza movimentos comuns de deslocamento em um único sentido. As cores das vestimentas também são diferenciadas do fundo, isto é, existem pontos com o contato direto entre o indivíduo e o fundo. A Figura 6.1 (d) apresenta um quadro que tipifica o Video4.avi.
- O Video5.avi apresenta uma pessoa em movimento em uma ambiente de fora de uma edificação, onde a pessoa realiza movimentos comuns de deslocamento em um único sentido. Nesse vídeos as cores das vestimentas são parcialmente diferenciadas do fundo, onde a pessoa passa por traz de uma pilastra e postes, a pilastra gera um oclusão completa da pessoa e os postes geram oclusões parciais. A Figura 6.1 (e) apresenta um quadro que tipifica o Video5.avi.
- O Video6.avi apresenta uma pessoa em movimento em uma sala, onde a pessoa realiza movimentos comuns de deslocamento em um único sentido, as cores das vestimentas são parcialmente diferenciadas do fundo. Próximo ao final do vídeo a pessoa passa por traz de uma televisão, gerando uma oclusão completa da pessoa. A Figura 6.1 (f) apresenta um quadro que tipifica o Video6.avi.

Tabela 6.1: Características dos vídeos utilizados para os testes.

Nome	Número de Quadros	Tempo (em segundos)	FPS	Resolução
Video1.avi	330	11	30	640 × 480
Video2.avi	240	8	30	640 × 480
Video3.avi	180	6	30	640 × 480
Video4.avi	120	4	30	640 × 480
Video5.avi	360	12	30	640 × 480
Video6.avi	390	13	30	640 × 480

6.1 Configurações dos Testes

Os vídeos utilizados, com exceção ao primeiro vídeo, obtido em [60], foram gravados em um ambiente controlado, buscando simular as características desejadas em relação ao ambiente, a resolução, a qualidade e o tempo. Conforme mencionado anteriormente, todos os vídeos possuem resolução 640×480 *pixels*, gravados em uma base fixas (sem movimentos no equipamento de captura), com um único alvo, e em ambientes abertos e fechado de edificações.



(a) Vídeo1.avi



(b) Vídeo2.avi



(c) Vídeo3.avi



(d) Vídeo4.avi



(e) Vídeo5.avi



(f) Vídeo6.avi

Figura 6.1: Quadros que tipificam os vídeos utilizados os testes.

Os vídeos apresentam diferenças nas cores do alvo, no tamanho do alvo, e na quantidade de quadros, sendo que esta última característica auxilia na identificação da persistência do método de rastreamento. A quantidade dos quadros utilizados não foi superior devido a necessidade de se realizar uma verificação periódica para identificar se não há um novo objeto no ambiente do vídeo.

Para se obter a máscara que segmenta o objeto alvo foi implementada uma ferramenta para capturar os quadros do vídeo, transformá-los em escala de cinza, compará-los sequencialmente e através do fluxo óptico, criar a máscara do quadro em formato binário. Para a obtenção do fluxo óptico foi utilizado o algoritmo de Horn-Schunck [58] com os seguintes parâmetros:

- Velocidades iniciais desativadas.
- Peso de suavidade com o valor $3 \cdot 10^{-5}$.

Foram testados os parâmetros listados abaixo com o objetivo de se identificar uma configuração que executasse o algoritmo de rastreamento de uma forma que se consumisse o mínimo de recurso computacional. Ou seja, os testes foram realizados de forma a executar o rastreamento sem onerar de forma agravante o desempenho do equipamento onde o algoritmo será executado. Portanto, os parâmetros testados foram:

- Quantidade de Partículas do PSO: n , sendo que o valor de n variou entre 100 e 200 partículas. Valor equivalente ao utilizado para o treinamento.
- Número de Iterações do PSO: n_{iter} , com valores que variaram entre 4 e 12.
- Blocos: testados com 8×8 pixels e 16×16 pixels para calcular histogramas RGB.
- Quantidade de Partículas para o Treinamento: $n \times 3$, onde $2n$ partículas são colocadas em posições fora do alvo.

Na execução do algoritmo de rastreamento proposto neste trabalho os parâmetros fixados em todos os testes foram:

- Grau de confiança em si, ou seja, na partícula melhor local, representado por $c1$: 2.05, $c1$ é apresentado na Equação 3.9.
- Grau de confiança no enxame, ou seja, na partícula melhor global, representado por $c2$: 2.05, $c2$ é apresentado na Equação 3.9.

O algoritmo proposto é heurístico e possui algumas variáveis, velocidades e distribuição das partículas, iniciadas de forma aleatória. Logo, se o algoritmo for executado inúmeras vezes ele pode apresentar resultados diferentes como resultado de cada uma dessas execuções. Nos testes foi levado em consideração a primeira execução com cada conjunto de parâmetros.

Os quadros que apresentam o alvo para rastreamento são os que apresentaram menor quantidade de ruídos nas máscaras, por não ser o foco da proposta do trabalho eles não foram abordados em detalhes, apesar de ter sido realizada uma estratégia de redução de ruído. A Tabela 6.2 identifica o primeiro quadro que é a posição de onde o rastreamento será inicializado para cada um dos vídeos nos testes do sistema de rastreamento proposto.

Tabela 6.2: Identificação do primeiro quadro de cada vídeo que será rastreado nos testes do sistema proposto.

Nome	Quadro Inicial
Video1.avi	51
Video2.avi	34
Video3.avi	40
Video4.avi	4
Video5.avi	142
Video6.avi	38

6.2 Resultados dos Testes com Sistema de Rastreamento

Os testes para cada vídeo são apresentados em duas etapas. Na primeira etapa são testadas três classes de parâmetros baseadas na quantidade de partículas (100, 150 e 200), sendo que em cada classe o número de iterações possui os valores 4, 8 e 12 e os blocos possuem as dimensões de 8×8 . Já na segunda etapa é avaliado o resultado da ampliação do tamanho do bloco que neste caso foi redimensionado para 16×16 . A Tabela 6.3 apresenta as relações de parâmetros dos testes realizados.

Tabela 6.3: Identificação das relações de parâmetro dos testes.

Quantidade de Partícula	Quantidade de Iterações
Classe 1: 100 Partículas	(a): 4 iterações
	(b): 8 iterações
	(c): 12 iterações
Classe 2: 150 Partículas	(a): 4 iterações
	(b): 8 iterações
	(c): 12 iterações
Classe 3: 200 Partículas	(a): 4 iterações
	(b): 8 iterações
	(c): 12 iterações

Portanto, para cada um dos vídeos foram executados 18 testes, sendo 9 para cada tamanho de bloco diferentes. Todas as tabelas relacionadas aos resultados dos testes para identificação do conjunto de parâmetros que melhor se adequam ao sistema de rastreamento proposto são apresentadas no Apêndice A e foram organizadas por vídeo. Estas tabelas apresentam o tempo (em segundos) que levou a execução do rastreamento, o número de quadros rastreados e se houve perda parcial ou total do alvo, ou não. Ao todo foram realizados 108 testes de rastreamento com o sistema proposto.

As Figuras 6.2, 6.3, 6.4, 6.5, 6.6 e 6.7 apresentam as partículas sobre o alvo na melhor relação de parâmetros com os blocos da função de custo de 16×16 no Video1.avi, Video2.avi, Video3.avi, Video4.avi, Video5.avi e Video6.avi, respectivamente.

Através da análise das tabelas apresentadas no Apêndice A foi possível gerar os gráficos, que sintetizam os resultados, apresentados a seguir. A Figura 6.8 apresenta a porcentagem de sucesso, rastreamento sem perda do alvo, em cada tamanho de bloco.

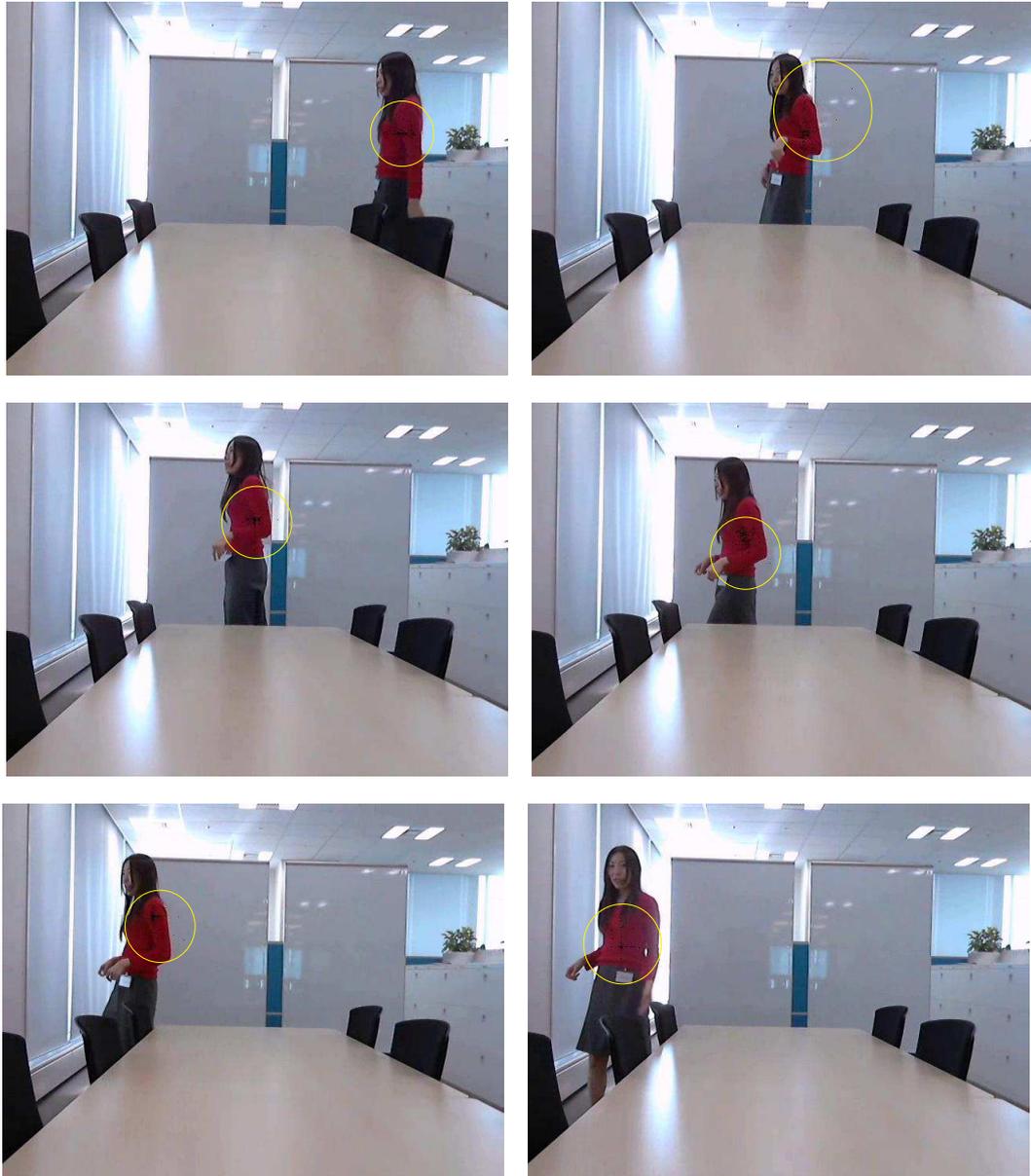


Figura 6.2: Quadros do rastreamento realizado no Video1.avi utilizando bloco 16×16 .

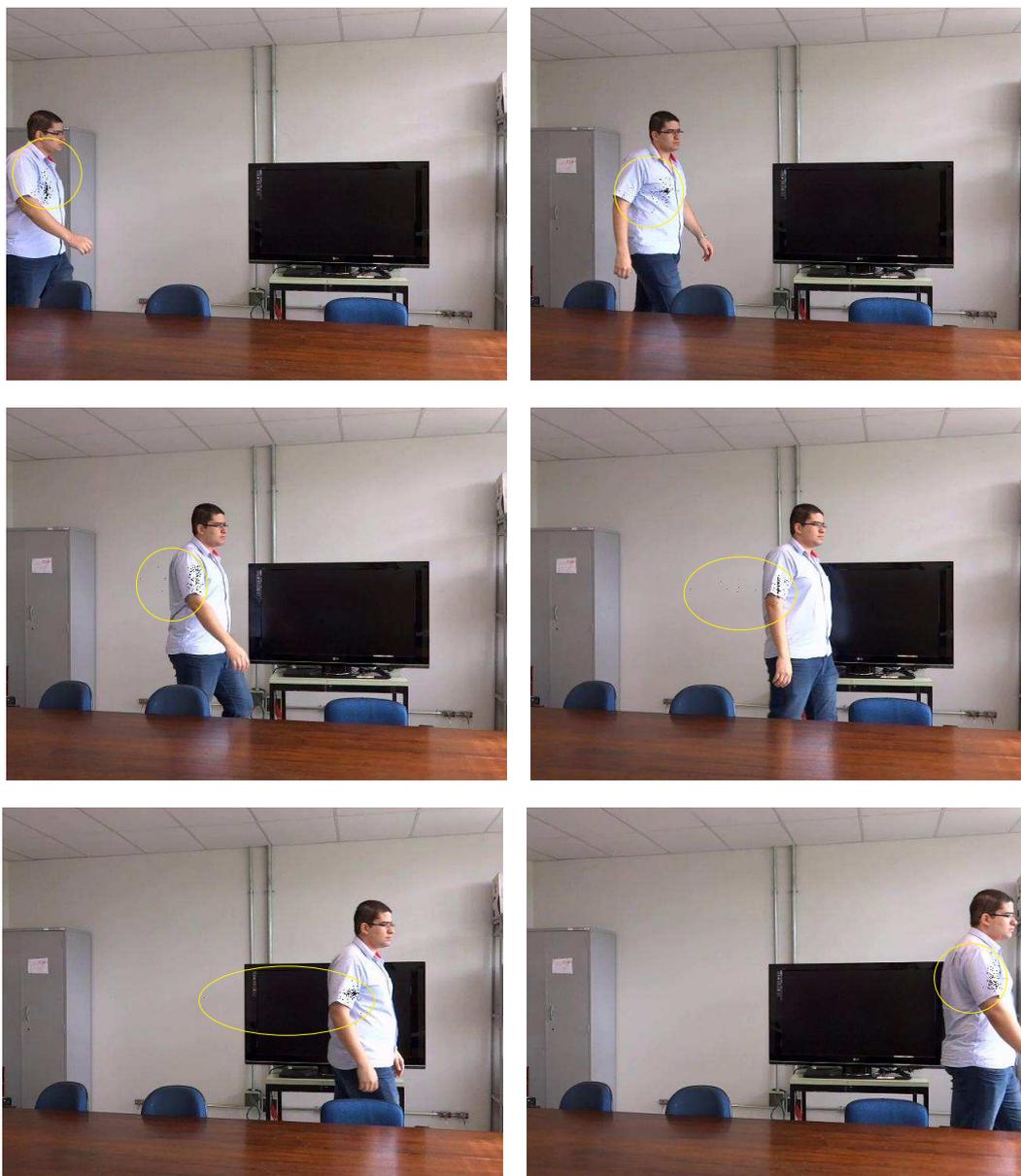


Figura 6.3: Quadros do rastreamento realizado no Video2.avi utilizando bloco 16×16 .



Figura 6.4: Quadros do rastreamento realizado no Video3.avi utilizando bloco 16×16 .



Figura 6.5: Quadros do rastreamento realizado no Video4.avi utilizando bloco 16×16 .

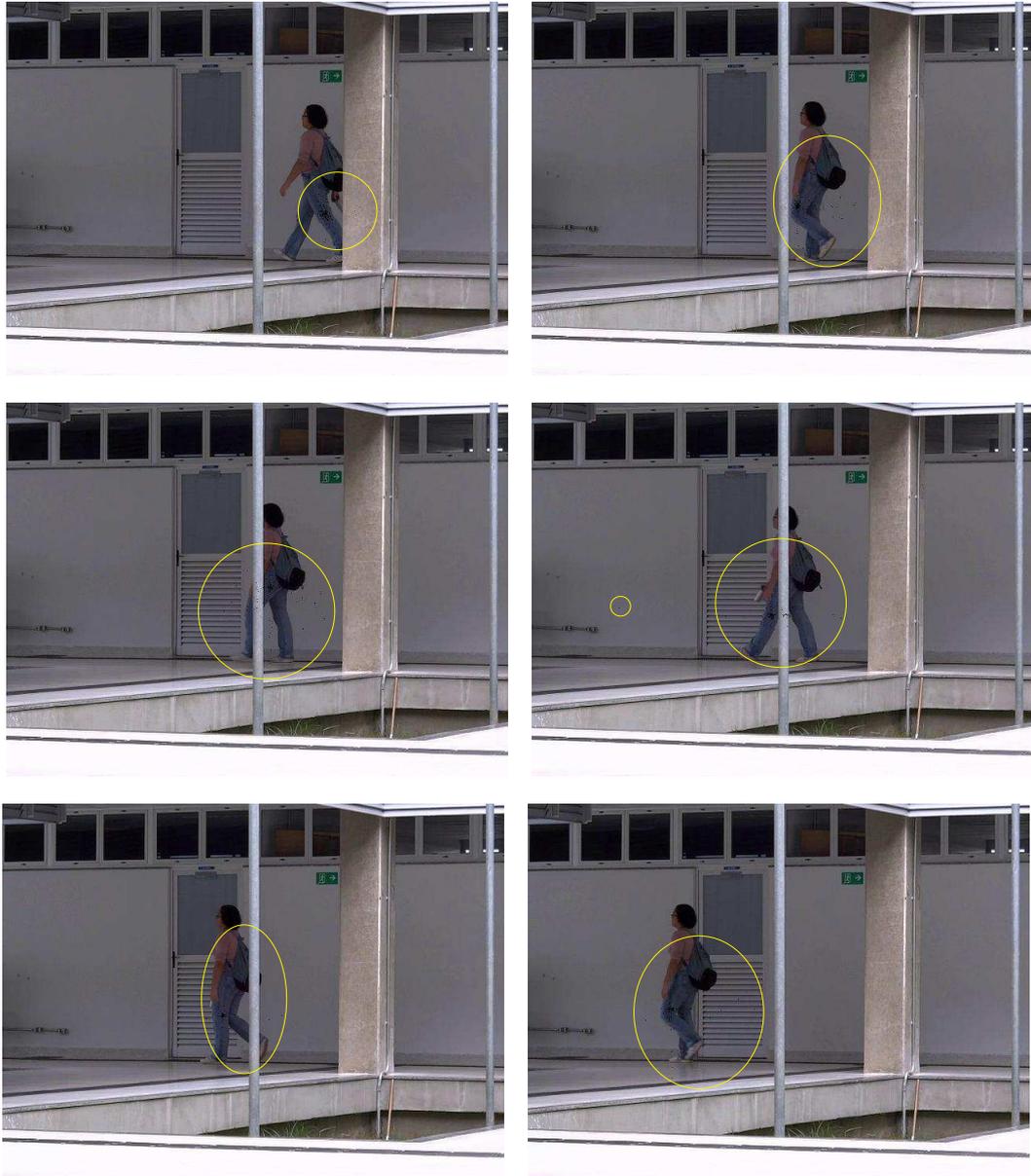


Figura 6.6: Quadros do rastreamento realizado no Video5.avi utilizando bloco 16×16 .

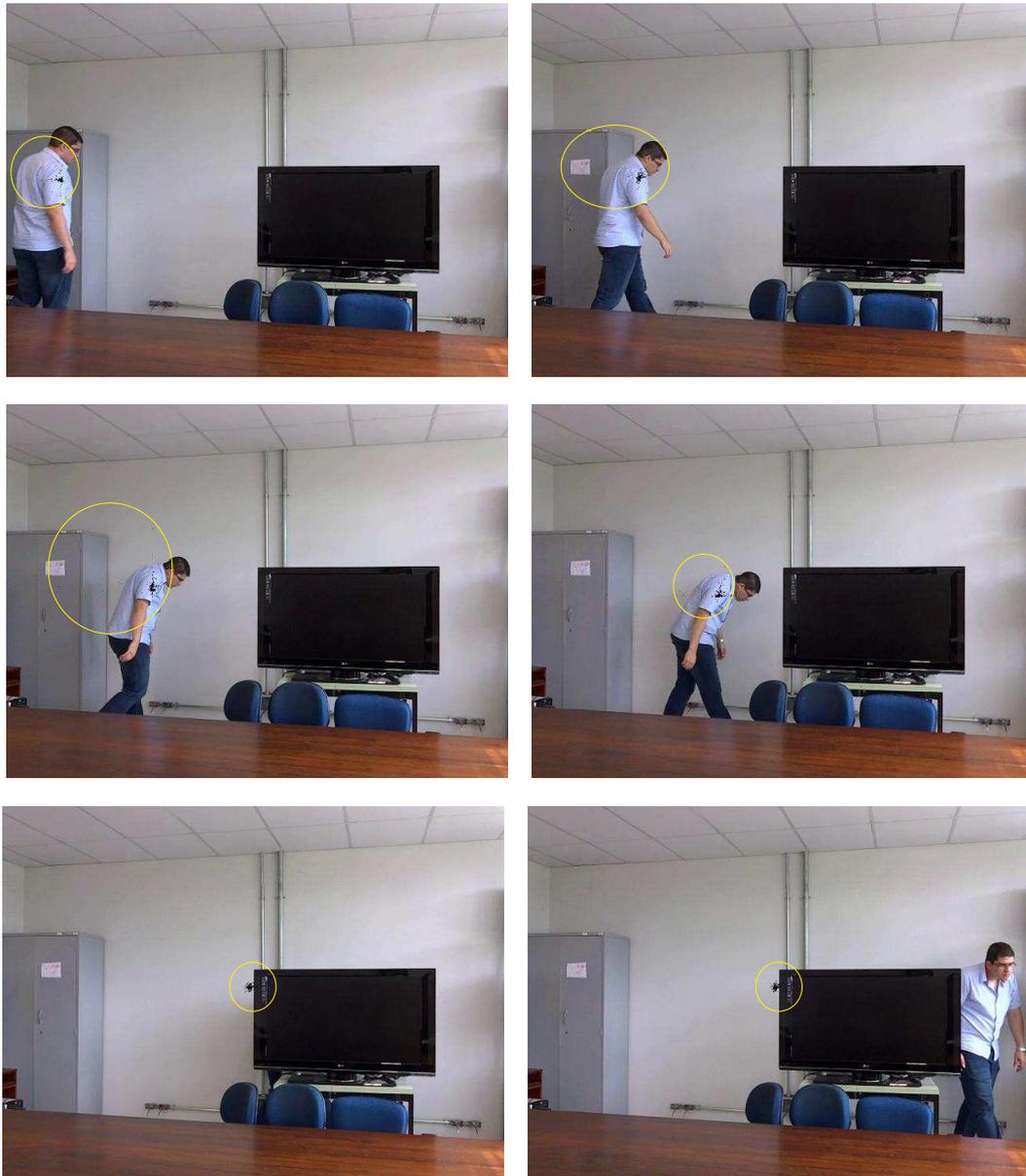


Figura 6.7: Quadros do rastreamento realizado no Video6.avi utilizando bloco 16×16 .

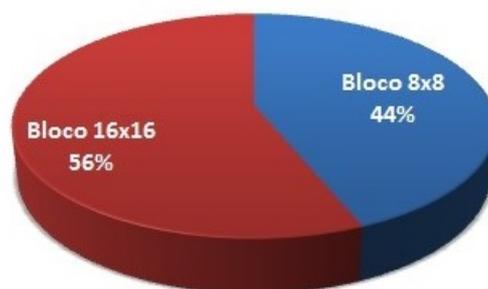


Figura 6.8: Porcentagem de sucesso no rastreamento em blocos 8×8 e 16×16 .

Dos 108 testes realizados 73 obtiveram sucesso, ou seja, rastreamento sem perda do alvo, sendo que 56% possuíam blocos de 16×16 e 44% blocos de 8×8 . A Figura 6.9 apresenta os gráficos de desempenho de tempo em CPU das relações que obtiveram sucesso no rastreamento utilizando blocos 8×8 e 16×16 .

Após a análise dos resultados apresentados na Figura 6.9, pode-se inferir que, genericamente, ou seja, independentemente do vídeo, o uso do bloco de 16×16 para a função de custo oferece maior estabilidade ao algoritmo como um todo. A queda de desempenho em termos de tempo, devido à adoção de blocos 16×16 é baixo, pois, na maioria das vezes não supera 1%, do tempo total.

A Figura 6.10 apresenta um gráfico com porcentagem de rastreamento com sucesso no bloco 16×16 distribuído por número de partículas. Assim é possível identificar qual o número de partículas que apresentaram melhores resultados durante o rastreamento utilizando bloco de 16×16 no método de PSO.

Com base no gráfico apresentado na Figura 6.10, é possível inferir que o número de partículas que ofereceu maior estabilidade durante o rastreamento proposto utilizando o bloco 16×16 , foi a classe que possuía 150 partículas, por ter obtido 37% dos rastreamentos com sucesso, seguido pela classe que possuía 100 partículas, por ter obtido 34% dos rastreamentos com sucesso.

Quando analisados os tempos consumidos na CPU, apresentados nas tabelas no Apêndice A, por cada classe com número de partículas diferentes durante o rastreamento com o sistema proposto, é possível identificar que o acréscimo de 50 partículas aumentam o tempo de rastreamento em cerca de 10% a 30%, independentemente do número de iterações. Portanto, a utilização do número de partículas entre 100 a 150 pode ser justificado visando o ganho de desempenho de tempo, uma vez que a diferença de sucesso através do uso de 100 partículas foi de apenas 3% perante o uso de 150 partículas.

A Figura 6.11 apresenta um gráfico com porcentagem de rastreamento com sucesso no bloco 16×16 distribuído por número de iterações. O objetivo deste gráfico é o de permitir a identificação de qual número de iterações que apresentou melhor resultado durante o rastreamento utilizando bloco de 16×16 no método de PSO.

Com base no gráfico apresentado na Figura 6.11, é possível inferir que o número de interações que ofereceram maior estabilidade durante o rastreamento proposto utilizando o bloco 16×16 , foram as classes que possuíam 8 e 12 iterações, ambas obtiveram 34% dos rastreamentos com sucesso e se diferenciaram em apenas 2% da classe com 4 iterações.

Ao analisar os tempos consumidos na CPU, apresentados nas tabelas no Apêndice A, por cada classe com número de iterações diferentes durante o rastreamento com o sistema proposto, é possível identificar que o acréscimo de 4 iterações aumentam o tempo de rastreamento em cerca de 25% a 50%, independentemente do número de partículas. Portanto, a utilização do número de iterações entre 4 a 8 pode ser justificado visando o ganho de desempenho de tempo, uma vez que a diferença de sucesso pode ser pequena.

6.3 Testes de Desempenho e Complexidade

Nesta seção é apresentada uma comparação do desempenho e complexidade do método determinístico de estimação de movimento com o estocástico baseado no PSO + LDF + Histogramas RGB, no momento após a detecção do alvo, utilizando os parâmetro que apresentaram resultados mais estáveis nos testes anteriores.

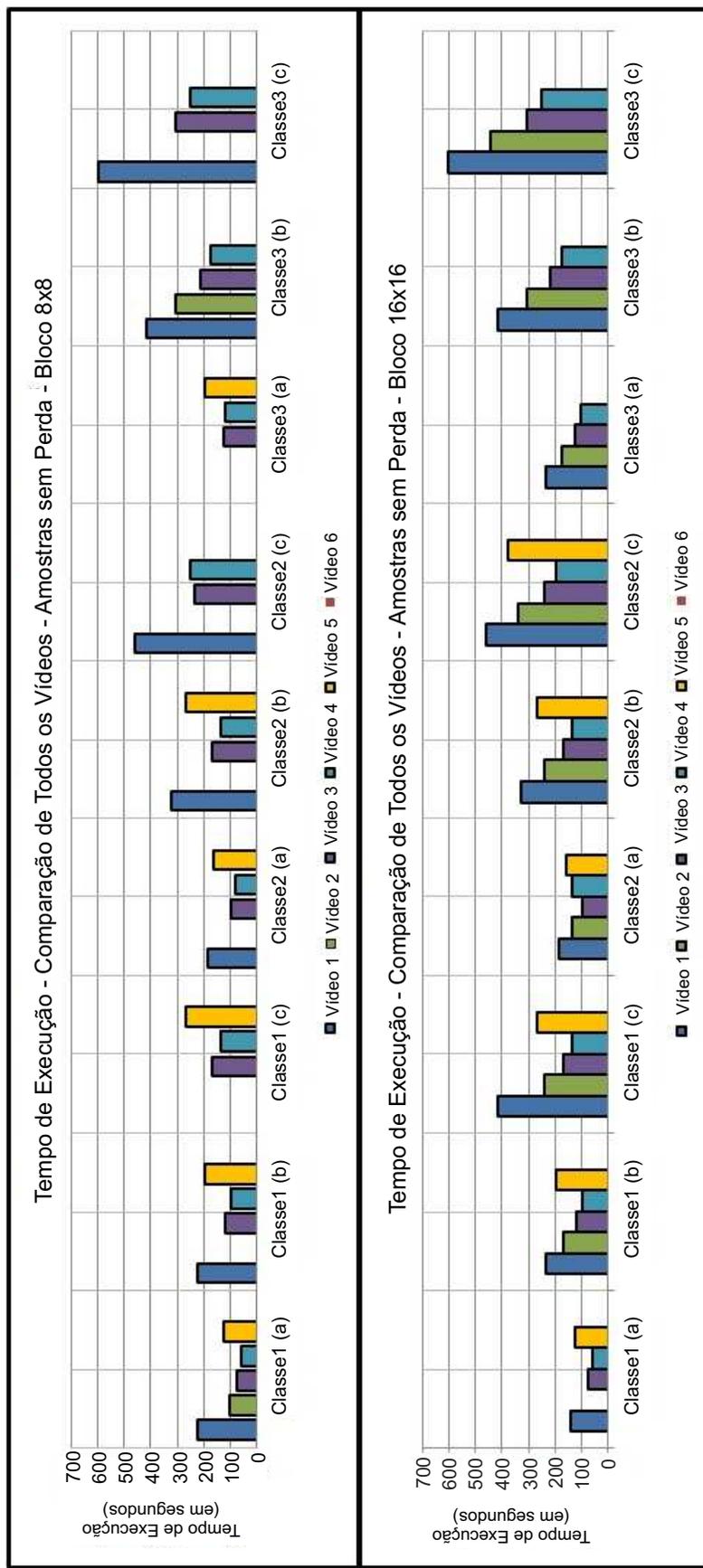


Figura 6.9: Gráfico que apresenta o desempenho de tempo em CPU das relações que obtiveram sucesso no rastreamento utilizando bloco 8×8 e 16×16 .

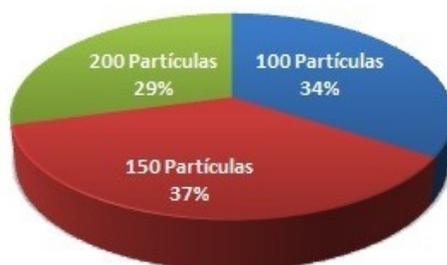


Figura 6.10: Porcentagem de rastreamento com sucesso no bloco 16×16 distribuído por número de partículas.



Figura 6.11: Porcentagem de rastreamento com sucesso no bloco 16×16 distribuído por número de iterações.

Foi implementado uma sistema de estimação de movimentos clássico [6], que recebe dois quadros sequenciais. Tal sistema tem como parâmetros, o tamanho do bloco e o tamanho da janela de busca, com os quais realizará o cálculo das diferenças entre os blocos para obter o vetor de movimentos. A Tabela 6.4 apresenta os resultados dos testes com o Video1.avi para os respectivos parâmetros no estimador de movimentos.

Tabela 6.4: Resultados dos testes com o Video1.avi para os determinados parâmetros no estimador de movimentos.

Tamanho do Bloco	Tamanho da Janela	Quantidade de Quadros	Tempo
8×8	32×32	1	474,73
8×8	32×32	10	4814,68
16×16	64×64	1	516,75
16×16	64×64	10	5240,30

O resultado do estimador de movimentos é preciso em relação aos blocos, porém, a demanda por recursos para processar tal sistema inviabiliza a utilização em aplicações de tempo real. Pois sua complexidade em cada quadro é de $O(n^2)$, obtida de $N^2 \times M^2 \times B$, onde N é o tamanho do bloco que será comparado, M é o tamanho da janela de busca e B é a quantidade de blocos do quadro.

Logo, visando a comparação com a estimação de movimentos foram definidos os parâmetros que oferecem estabilidade para o rastreador proposto. A Tabela 6.5 apresenta os resultados dos testes com o Video1.avi para os respectivos parâmetros no rastreador proposto.

Tabela 6.5: Resultados dos testes com o Video1.avi para os respectivos parâmetros no rastreador proposto.

Tamanho do Bloco	Núm. de Partícula	Núm. de Iterações	Quant. de Quadros	Tempo
16×16	150	8	1	15,38
16×16	150	8	10	25,64

O desempenho de tempo do algoritmo proposto possui maior desgaste no treinamento das funções de custo do PSO. Por ser um algoritmo estocástico com treinamento ele não possui um desempenho de tempo escalar. Porém é possível identificar que após o treinamento há um desempenho de tempo com poucas oscilações no algoritmo proposto. A complexidade assumida pelo do algoritmo proposto nas fases de treinamento e de rastreamento é $O(n)$, obtida de $3 \times N \times B + N \times I \times B$, onde N é o número de partículas, B é a quantidade de blocos do quadro para obtenção do histograma RGB e I é o número de iterações do PSO.

A Figura 6.12 apresenta o gráfico com a comparação do desempenho em termos de tempo para a execução do estimador de movimentos com o método de rastreamento proposto (PSO + LDF + Histogramas RGB) no decorrer da leitura das sequências de imagens.

6.4 Testes de Rastreamento do Algoritmo Proposto

O algoritmo proposto, prevê a reinicialização do processo para obtenção do alvo através de estimação de movimentos perante a situação da perda do mesmo durante o rastreamento. Anteriormente, os testes não previam tal reinicialização por estar avaliando as características de

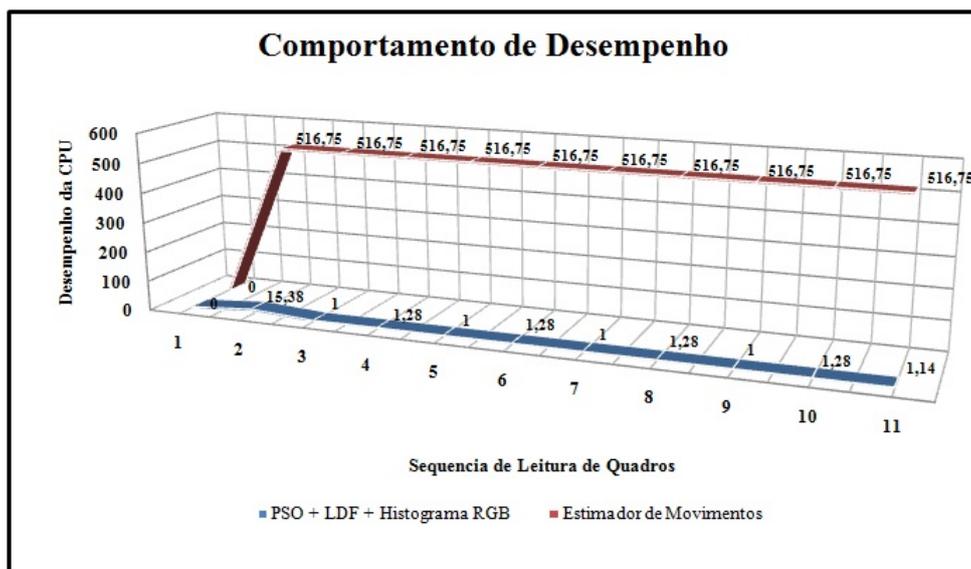


Figura 6.12: Gráfico que apresenta o comportamento de execução do estimador de movimentos e da proposta aqui apresentada (PSO + LDF + Histogramas RGB).

diversos parâmetro, sendo necessária a verificação da capacidade de recuperação do rastreamento por si próprio.

Caso fossem utilizados os parâmetros que melhores se adequaram ao algoritmo de rastreamento, possivelmente, o rastreador iria até o fim dos quadros do vídeo sem a perda do alvo em nenhum quadro. Portanto, para o teste de recuperação do alvo via a máscara de segmentação do mesmo, obtida através do processo de estimação de movimento, é que para o Video2.avi foram utilizados os seguintes parâmetros:

- Bloco para obtenção do histograma: 8×8 .
- Quantidade de partículas: 100.
- Quantidade de iterações: 8.

Tais parâmetro são iguais aos apresentados na Classe 1 (b) da Tabela A.4. A execução do rastreamento do alvo foi iniciada no quadro 34 do Video2.avi, no quadro 74 o alvo foi perdido, sendo necessária nova busca pelo alvo através da máscara com a segmentação do alvo e um novo treinamento. Em seguida, o rastreamento pelo PSO foi continuado, obtendo sucesso no rastreamento até o quadro 232, próximo ao desaparecimento do alvo, no vídeo. Após a busca do alvo via estimação de movimento e treinamento da função de custo, o rastreamento pelo PSO foi continuado até que o alvo desaparecesse totalmente do vídeo.

A Figura 6.13 apresenta os quadros do processo de rastreamento do teste do algoritmo proposto.

Após a realização do teste foi possível identificar que tal método possibilita a estruturação do rastreamento visual com grande redução do consumo de recursos computacionais sem perder a capacidade de rastreamento do alvo. Em alguns sistemas pode-se utilizar o método proposto com parâmetros menos custosos e fazer uso de uma estratégia de reinicialização, que faz a estimação de movimento e o treinamento da função de custo, de forma contínua a cada ciclo de tempo. Para isto, basta apenas criar uma verificação da posição do quadro lido.

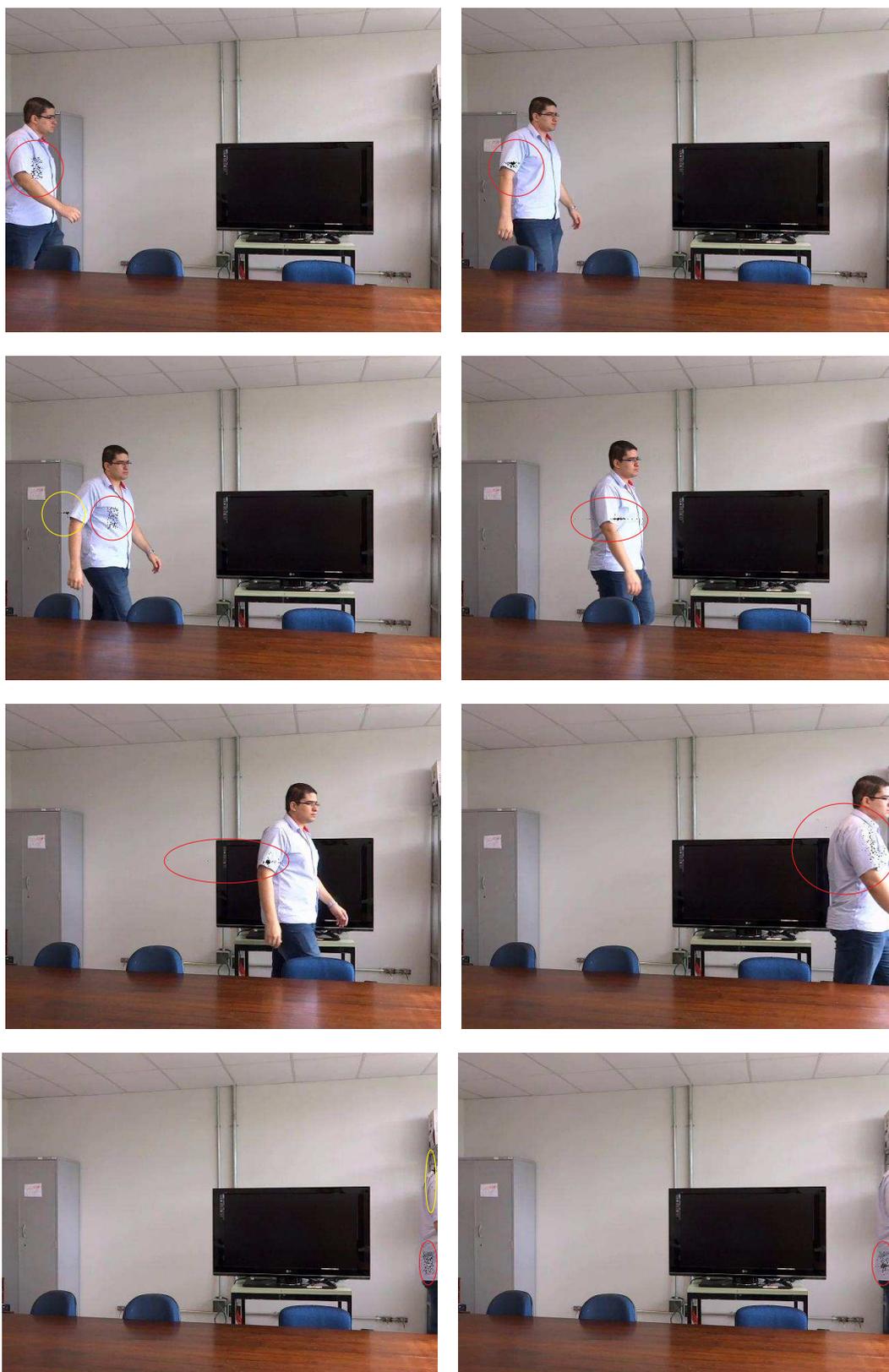


Figura 6.13: Quadros do teste de rastreamento utilizando o algoritmo completo, onde no primeiro, terceiro e no penúltimo quadro é realizado o processo de estimação de movimentos.

6.5 Teste de Rastreamento de Vídeo com Oclusão

O Video5.avi apresenta durante a movimentação do alvo oclusão total e oclusão parcial do mesmo, que se desloca em um sentido único, muitas técnicas de rastreamento encontram dificuldades perante as oclusões totais do alvo. A seguir é apresentado um teste de rastreamento no Video5.avi utilizando o algoritmo proposto com uma pequena adaptação. A adaptação consiste na ampliação gradativa da janela de busca do método de PSO. Geralmente um vídeo apresenta cerca de 25 a 30 quadros por segundo, portanto espera-se que o deslocamento do alvo seja pequeno. Baseado nesta característica o método proposto, apresentado anteriormente, utilizou um tamanho de janela de busca na reinicialização das partículas do PSO de 32×32 pixels ao redor da partícula melhor global.

Em se tratando do PSO, tamanho da janela de busca pode ser vantajosa em algumas situações, dentre elas situações com oclusões e quando há movimentos abruptos, pois torna possível monitorar uma área maior do vídeo. Porém, como desvantagem tem-se a perda de desempenho dos recursos computacionais, caso seja adicionada maior quantidade de partículas, ou tem-se a perda de capacidade de rastreamento, caso não seja adicionado mais partículas ou exista outro objeto com as mesmas características do alvo em outra região da imagem.

Para implementar a adaptação do algoritmo proposto comentada anteriormente foi necessário aumentar a tolerância da perda do alvo, pois em um primeiro momento ao rastreador, perder o alvo, ou seja, obter um baixo valor na função de custo da melhor posição global, era demandada a imediata obtenção do alvo via estimação de movimento, exigindo novo treinamento da função de custo. A tolerância implementada foi que somente após o quarto quadro perdido consecutivamente era realizada a obtenção do alvo via estimação de movimento e novo treinamento da função de custo.

A adaptação contou com ampliação do tamanho da janela de busca através de uma multiplicação por 3 após a primeira perda do alvo e na segunda perda do alvo o tamanho da janela de busca original foi multiplicada por 5. Para tal resolução, nos testes realizados, não foi pretendido um tamanho de janela de busca superior a 160×160 , devido a característica de redução da capacidade de rastreamento com janelas de busca de tamanhos superiores. A Tabela 6.6 apresenta os parâmetros adotados no teste com janela de busca ampliada gradativamente no algoritmo proposto.

Tabela 6.6: Parâmetros adotados no teste com janela de busca ampliada gradativamente no algoritmo proposto no Video5.avi.

Quantidade de Partícula	Quantidade de Iterações	Tamanho do Bloco	Quadro Inicial
150 partículas	8 iterações	16×16 pixels	64

A Figura 6.14 apresenta os quadros do rastreamento realizado no teste do rastreamento com o algoritmo proposto com a janela de busca ampliada gradativamente no Video5.avi.



Figura 6.14: Quadros do teste de rastreamento utilizando o algoritmo proposto com a estratégia para rastrear alvos com oclusão.

Capítulo 7

Considerações Finais

Neste trabalho foram abordados conceitos dos principais métodos de rastreamentos, dentre eles, o método baseado na Otimização por Enxame de Partículas, que recentemente tem demandado muitos trabalhos de pesquisa, geralmente, motivados pela criação de sistemas de rastreamento multi-alvos e rastreamento em tempo-real. Um dos pontos essenciais do método de Otimização por Enxame de Partículas é a refinada caracterização do objeto alvo durante o rastreamento visual através de sua função de custo.

Desta forma, foi proposto um algoritmo baseado em uma técnica de classificação supervisionada, utilizando Funções Lineares Discriminante, para fazer uso de histogramas RGB de blocos dos quadros para caracterização do objeto alvo, isto, através da função de custo do modelo estocástico de rastreamento visual, gerado a partir do método de Otimização por Enxame de Partículas.

No algoritmo proposto, após o segundo quadro capturado, seja de equipamento de captura de vídeo ou de um arquivo de vídeo, realiza-se a detecção do objeto alvo através da estimação de movimento e obtenção do fluxo óptico. Tal processo permite a realização do treinamento da função de custo do método de PSO utilizando a LDF. Logo em seguida, é chamado um novo quadro que terá o alvo rastreado através do método de PSO, que se baseia em três elementos essenciais, a inércia, melhor posição local e melhor posição global, para atualizar o deslocamento do enxame e acompanhar o alvo. As atualizações da melhor posição local e melhor posição global é definida pela comparação da função de custo, que quanto mais próxima de '1' indica que a posição está mais próxima ou sobre o alvo.

Através da execução do conjunto de testes, foi possível a identificação de várias características do algoritmo proposto, dentre elas, foi identificado que histogramas RGB de blocos de 16×16 pixels, agregados com um conjunto de 100 - 150 partículas, e cerca de 4 - 12 iterações oferecem boa estabilidade de rastreamento de indivíduos em vídeos de resolução 640×480 pixels utilizando equipamentos de captura estáticos. Na identificação dos parâmetros para o algoritmo proposto é que os histogramas RGB de blocos com dimensões de 16×16 pixels ofereceram maior estabilidade do que histogramas RGB de blocos de dimensões de 8×8 pixels agregando perda de desempenho quase que insignificante.

Os resultados dos testes de comparação de desempenho do rastreamento proposto com o método de rastreamento por estimação de movimento demonstraram que é possível obter um ganho de desempenho significativo nos sistemas de rastreamento através da utilização do rastreamento utilizando o PSO com função de custo LDF e com histogramas RGB. Porém, é importante ressaltar que o algoritmo proposto completo faz uso da estimação de movimento

para a obtenção do alvo, para que seja viável o treinamento da função de custo baseado no LDF com histogramas RGB de blocos, sobre o alvo.

Através de testes foi apresentada uma estratégia para tratamento de oclusões do objeto alvo, sendo que tal estratégia consistiu na ampliação gradativa da janela de busca do enxame de partículas e maior tolerância a perda do alvo pelo algoritmo de rastreamento. O resultado dos testes da estratégia demonstrou boa capacidade de superação de oclusões parciais e totais do alvo, oferecendo maior robustez ao algoritmo proposto.

Através dos testes realizados é possível identificar que o algoritmo proposto oferece rápida convergência, pois foram obtidos bons resultados com poucas iterações no PSO; baixo custo computacional, se comparado com métodos determinísticos comuns; boa capacidade de tratar oclusões que não superam 1/4 (um quarto) da resolução do vídeo; e boa capacidade de rastreamento de objeto com movimentos arbitrários e abruptos.

Em trabalhos futuros, diversos mecanismos podem ser incorporados ao algoritmo proposto, objetivando a ampliação da aplicação e melhoria no funcionamento do mesmo. Uma possível melhoria é adequação do algoritmo de rastreamento para detecção de múltiplos objetos como alvo do rastreamento, pois é muito comum a existência de múltiplos alvos em vídeos de monitoramento. Além, da utilização de mecanismos para obtenção dos objetos alvos com menor custo computacional, visando melhorar ainda mais o desempenho do algoritmo proposto, e alteração do sistema de cores RGB para HSV, visando reforçar ainda mais a caracterização do objeto alvo.

Apêndice A

Tabelas com os Resultados dos Testes com Sistema de Rastreamento

A Tabela A.1 identifica as relações de parâmetros dos testes realizados.

Tabela A.1: Identificação das relações de parâmetro dos testes.

Quantidade de Partícula	Quantidade de Iterações
Classe 1: 100 Partículas	(a): 4 iterações
	(b): 8 iterações
	(c): 12 iterações
Classe 2: 150 Partículas	(a): 4 iterações
	(b): 8 iterações
	(c): 12 iterações
Classe 3: 200 Partículas	(a): 4 iterações
	(b): 8 iterações
	(c): 12 iterações

A.0.1 Vídeo 1

A Tabela A.2 apresenta os resultados obtidos com a adoção dos parâmetros apresentados na Tabela A.1, utilizando o algoritmo de rastreamento proposto com os blocos da função de custo com dimensões equivalentes a 8×8 para o Video1.avi. Na Tabela A.2 a perda total é quando o rastreador perde o alvo e não consegue se recuperar, a perda parcial é quando o rastreador perde o alvo, porém o rastreador consegue se recuperar e sem perca é quando se esgotou os quadros do vídeo e o rastreador em nenhum quadro perdeu alvo.

Tabela A.2: Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo com dimensões de 8×8 - Video1.avi.

Parâmetros	Tempo (s)	Quadros Rastreados	Tipo de Perca	Porcentagem de Quadros Rastreados
Classe 1 (a)	137,74	157	Total	56%
Classe 1 (b)	226,55	279	Sem Perca	100%
Classe 1 (c)	321,42	155	Total	55%
Classe 2 (a)	186,66	279	Sem Perca	100%
Classe 2 (b)	323,72	279	Sem Perca	100%
Classe 2 (c)	460,33	279	Sem Perca	100%
Classe 3 (a)	233,76	157	Total	56%
Classe 3 (b)	413,67	279	Sem Perca	100%
Classe 3 (c)	598,06	279	Sem Perca	100%

A Tabela A.3 apresenta os resultados da adoção dos parâmetros demonstrados em A.1 no algoritmo de rastreamento proposto com os blocos da função de custo com dimensão de 16×16 para o Video1.avi. Na Tabela A.3 a perda total é quando o rastreador perde o alvo e não consegue se recuperar, a perda parcial é quando o rastreador perde o alvo, porém o rastreador consegue se recuperar e sem perca é quando se esgotou os quadros do vídeo e o rastreador em nenhum quadro perdeu alvo.

Tabela A.3: Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 16×16 para o Video1.avi.

Parâmetros	Tempo (s)	Quadros Rastreados	Tipo de Perca	Porcentagem de Quadros Rastreados
Classe 1 (a)	139,87	279	Sem Perca	100%
Classe 1 (b)	232,32	279	Sem Perca	100%
Classe 1 (c)	415,43	279	Sem Perca	100%
Classe 2 (a)	186,39	279	Sem Perca	100%
Classe 2 (b)	325,60	279	Sem Perca	100%
Classe 2 (c)	461,03	279	Sem Perca	100%
Classe 3 (a)	233,20	279	Sem Perca	100%
Classe 3 (b)	417,57	279	Sem Perca	100%
Classe 3 (c)	602,48	279	Sem Perca	100%

A.0.2 Vídeo 2

A Tabela A.4 apresenta os resultados da adoção dos parâmetros demonstrados na Tabela A.1 para a execução do algoritmo de rastreamento proposto com os blocos da função de custo com dimensões de 8×8 para o Video2.avi. Na Tabela A.4 a perda total é quando o rastreador perde o alvo e não consegue se recuperar, a perda parcial é quando o rastreador perde o alvo, porém o rastreador consegue se recuperar e sem perca é quando se esgotou os quadros do vídeo e o rastreador em nenhum quadro perdeu alvo.

Tabela A.4: Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 8×8 para o Video2.avi.

Parâmetros	Tempo (s)	Quadros Rastreados	Tipo de Perca	Porcentagem de Quadros Rastreados
Classe 1 (a)	103,81	206	Sem Perca	100%
Classe 1 (b)	172,34	54	Total	26%
Classe 1 (c)	238,16	56	Total	27%
Classe 2 (a)	138,22	64	Total	31%
Classe 2 (b)	238,92	50	Total	24%
Classe 2 (c)	339,68	94	Total	45%
Classe 3 (a)	172,63	100	Total	48%
Classe 3 (b)	306,59	206	Sem Perca	100%
Classe 3 (c)	440,41	68	Total	33%

A Tabela A.5 apresenta os resultados da adoção dos parâmetros demonstrados na Tabela A.1 para o algoritmo de rastreamento proposto, utilizando os blocos da função de custo com dimensões de 16×16 para o Video2.avi. Na Tabela A.5 a perda total é quando o rastreador perde o alvo e não consegue se recuperar, a perda parcial é quando o rastreador perde o alvo, porém o rastreador consegue se recuperar e sem perca é quando se esgotou os quadros do vídeo e o rastreador em nenhum quadro perdeu alvo.

Tabela A.5: Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 16×16 para o Video2.avi.

Parâmetros	Tempo (s)	Quadros Rastreados	Tipo de Perca	Porcentagem de Quadros Rastreados
Classe 1 (a)	104,90	148	Total	71%
Classe 1 (b)	170,87	206	Sem Perca	100%
Classe 1 (c)	238,92	206	Sem Perca	100%
Classe 2 (a)	138,40	206	Sem Perca	100%
Classe 2 (b)	242,33	206	Sem Perca	100%
Classe 2 (c)	340,81	206	Sem Perca	100%
Classe 3 (a)	173,27	206	Sem Perca	100%
Classe 3 (b)	307,91	206	Sem Perca	100%
Classe 3 (c)	444,45	206	Sem Perca	100%

A.0.3 Vídeo 3

A Tabela A.6 apresenta os resultados da adoção dos parâmetros demonstrados em A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de dimensão de 8×8 para o Video3.avi. Na Tabela A.6 a perda total é quando o rastreador perde o alvo e não consegue se recuperar, a perda parcial é quando o rastreador perde o alvo, porém o rastreador consegue se recuperar e sem perca é quando se esgotou os quadros do vídeo e o rastreador em nenhum quadro perdeu alvo.

Tabela A.6: Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 8×8 para o Video3.avi.

Parâmetros	Tempo (s)	Quadros Rastreados	Tipo de Perca	Porcentagem de Quadros Rastreados
Classe 1 (a)	75,22	140	Sem Perca	100%
Classe 1 (b)	121,46	140	Sem Perca	100%
Classe 1 (c)	167,53	140	Sem Perca	100%
Classe 2 (a)	98,84	140	Sem Perca	100%
Classe 2 (b)	167,79	140	Sem Perca	100%
Classe 2 (c)	236,59	140	Sem Perca	100%
Classe 3 (a)	122,47	140	Sem Perca	100%
Classe 3 (b)	214,50	140	Sem Perca	100%
Classe 3 (c)	306,92	140	Sem Perca	100%

A Tabela A.7 apresenta os resultados da adoção dos parâmetros demonstrados na Tabela A.1 para o algoritmo de rastreamento proposto com os blocos da função de custo com dimensões iguais a 16×16 para o Video3.avi. Na Tabela A.7 a perda total é quando o rastreador perde o alvo e não consegue se recuperar, a perda parcial é quando o rastreador perde o alvo, porém o rastreador consegue se recuperar e sem perca é quando se esgotou os quadros do vídeo e o rastreador em nenhum quadro perdeu alvo.

Tabela A.7: Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 16×16 para o Video3.avi.

Parâmetros	Tempo (s)	Quadros Rastreados	Tipo de Perca	Porcentagem de Quadros Rastreados
Classe 1 (a)	75,35	140	Sem Perca	100%
Classe 1 (b)	121,91	140	Sem Perca	100%
Classe 1 (c)	167,95	140	Sem Perca	100%
Classe 2 (a)	99,19	140	Sem Perca	100%
Classe 2 (b)	168,80	140	Sem Perca	100%
Classe 2 (c)	237,83	140	Sem Perca	100%
Classe 3 (a)	123,03	140	Sem Perca	100%
Classe 3 (b)	215,75	140	Sem Perca	100%
Classe 3 (c)	307,79	140	Sem Perca	100%

A.0.4 Vídeo 4

A Tabela A.8 apresenta os resultados da adoção dos parâmetros demonstrados na Tabela A.1 para o algoritmo de rastreamento proposto com os blocos da função de custo de dimensões equivalentes a 8×8 para o Video4.avi. Na Tabela A.8 a perda total é quando o rastreador perde o alvo e não consegue se recuperar, a perda parcial é quando o rastreador perde o alvo, porém o rastreador consegue se recuperar e sem perca é quando se esgotou os quadros do vídeo e o rastreador em nenhum quadro perdeu alvo.

Tabela A.8: Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 8×8 para o Video4.avi.

Parâmetros	Tempo (s)	Quadros Rastreados	Tipo de Perca	Porcentagem de Quadros Rastreados
Classe 1 (a)	60,37	116	Sem Perca	100%
Classe 1 (b)	98,86	116	Sem Perca	100%
Classe 1 (c)	136,96	116	Sem Perca	100%
Classe 2 (a)	79,91	116	Sem Perca	100%
Classe 2 (b)	137,26	116	Sem Perca	100%
Classe 2 (c)	252,07	116	Sem Perca	100%
Classe 3 (a)	121,21	116	Sem Perca	100%
Classe 3 (b)	175,76	116	Sem Perca	100%
Classe 3 (c)	251,07	116	Sem Perca	100%

A Tabela A.9 apresenta os resultados da adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de dimensões iguais a 16×16 para o Video4.avi. Na Tabela A.9 a perda total é quando o rastreador perde o alvo e não consegue se recuperar, a perda parcial é quando o rastreador perde o alvo, porém o rastreador consegue se recuperar e sem perca é quando se esgotou os quadros do vídeo e o rastreador em nenhum quadro perdeu alvo.

Tabela A.9: Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 16×16 para o Video4.avi.

Parâmetros	Tempo (s)	Quadros Rastreados	Tipo de Perca	Porcentagem de Quadros Rastreados
Classe 1 (a)	60,20	116	Sem Perca	100%
Classe 1 (b)	98,41	116	Sem Perca	100%
Classe 1 (c)	137,27	116	Sem Perca	100%
Classe 2 (a)	79,95	116	Sem Perca	100%
Classe 2 (b)	137,83	116	Sem Perca	100%
Classe 2 (c)	194,42	116	Sem Perca	100%
Classe 3 (a)	100,77	116	Sem Perca	100%
Classe 3 (b)	176,79	116	Sem Perca	100%
Classe 3 (c)	253,59	116	Sem Perca	100%

A.0.5 Vídeo 5

A Tabela A.10 apresenta os resultados da adoção dos parâmetros demonstrados na Tabela A.1 para o algoritmo de rastreamento proposto com os blocos da função de custo com dimensões de 8×8 para o Video5.avi. Na Tabela A.10 a perda total é quando o rastreador perde o alvo e não consegue se recuperar, a perda parcial é quando o rastreador perde o alvo, porém o rastreador consegue se recuperar e sem perca é quando se esgotou os quadros do vídeo e o rastreador em nenhum quadro perdeu alvo.

Tabela A.10: Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 8×8 para o Video5.avi.

Parâmetros	Tempo (s)	Quadros Rastreados	Tipo de Perca	Porcentagem de Quadros Rastreados
Classe 1 (a)	125,46	218	Sem Perca	100%
Classe 1 (b)	195,13	218	Sem Perca	100%
Classe 1 (c)	267,17	218	Sem Perca	100%
Classe 2 (a)	163,05	218	Sem Perca	100%
Classe 2 (b)	266,78	218	Sem Perca	100%
Classe 2 (c)	375,05	217	Parcial	99%
Classe 3 (a)	196,81	218	Sem Perca	100%
Classe 3 (b)	340,03	211	Total	96%
Classe 3 (c)	482,68	217	Parcial	99%

A Tabela A.11 apresenta os resultados da adoção dos parâmetros demonstrados na Tabela A.1 para o algoritmo de rastreamento proposto com os blocos da função de custo com dimensões iguais a 16×16 para o Video5.avi. Na Tabela A.11 a perda total é quando o rastreador perde o alvo e não consegue se recuperar, a perda parcial é quando o rastreador perde o alvo, porém o rastreador consegue se recuperar e sem perca é quando se esgotou os quadros do vídeo e o rastreador em nenhum quadro perdeu alvo.

Tabela A.11: Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 16×16 para o Video5.avi.

Parâmetros	Tempo (s)	Quadros Rastreados	Tipo de Perca	Porcentagem de Quadros Rastreados
Classe 1 (a)	124,19	218	Sem Perca	100%
Classe 1 (b)	196,34	218	Sem Perca	100%
Classe 1 (c)	268,23	218	Sem Perca	100%
Classe 2 (a)	160,27	218	Sem Perca	100%
Classe 2 (b)	267,86	218	Sem Perca	100%
Classe 2 (c)	376,18	218	Sem Perca	100%
Classe 3 (a)	197,85	47	Total	21%
Classe 3 (b)	341,33	42	Total	19%
Classe 3 (c)	484,44	217	Parcial	99%

A.0.6 Vídeo 6

A Tabela A.12 apresenta os resultados da adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo com dimensões de 8×8 para o Video6.avi. Na Tabela A.12 a perda total é quando o rastreador perde o alvo e não consegue se recuperar, a perda parcial é quando o rastreador perde o alvo, porém o rastreador consegue se recuperar e sem perca é quando se esgotou os quadros do vídeo e o rastreador em nenhum quadro perdeu alvo.

Tabela A.12: Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 8×8 para o Video6.avi.

Parâmetros	Tempo (s)	Quadros Rastreados	Tipo de Perca	Porcentagem de Quadros Rastreados
Classe 1 (a)	172,69	152	Total	43%
Classe 1 (b)	287,06	174	Total	49%
Classe 1 (c)	402,04	178	Total	50%
Classe 2 (a)	229,65	106	Total	30%
Classe 2 (b)	403,30	173	Total	49%
Classe 2 (c)	575,05	88	Total	25%
Classe 3 (a)	290,81	179	Total	50%
Classe 3 (b)	519,62	106	Total	30%
Classe 3 (c)	575,00	111	Total	31%

A Tabela A.13 apresenta os resultados da adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 16×16 para o Video6.avi. Na Tabela A.13 a perda total é quando o rastreador perde o alvo e não consegue se recuperar, a perda parcial é quando o rastreador perde o alvo, porém o rastreador consegue se recuperar e sem perca é quando se esgotou os quadros do vídeo e o rastreador em nenhum quadro perdeu alvo.

Tabela A.13: Adoção dos parâmetros demonstrados na Tabela A.1 no algoritmo de rastreamento proposto com os blocos da função de custo de 16×16 para o Video6.avi.

Parâmetros	Tempo (s)	Quadros Rastreados	Tipo de Perca	Porcentagem de Quadros Rastreados
Classe 1 (a)	173,38	179	Total	50%
Classe 1 (b)	288,88	176	Total	50%
Classe 1 (c)	404,56	174	Total	49%
Classe 2 (a)	229,98	177	Total	50%
Classe 2 (b)	402,30	179	Total	50%
Classe 2 (c)	573,90	179	Total	50%
Classe 3 (a)	292,11	169	Total	48%
Classe 3 (b)	521,32	161	Total	45%
Classe 3 (c)	573,54	164	Total	46%

Referências

- [1] C. Hsu and G. Dai. Multiple object tracking using particle swarm optimization. In *World Academy of Science, Engineering and Technology, 2012, issue 68*, pages 41–44, 2012. 2, 35
- [2] X. Zhang, W. Hu, S. Maybank, X. Li, and M. Zhu. Sequential particle swarm optimization for visual tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008. 2, 35
- [3] Y. Zheng and Y. Meng. Swarm intelligence based dynamic object tracking. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 405–412, 2008. 2, 35
- [4] V. John, E. Trucco, and S. Ivekovic. Markerless human articulated tracking using hierarchical particle swarm optimisation. *Image and Vision Computing*, 28(11):1530 – 1547, 2010. 2, 35
- [5] T. Wiegand, J.-R. Ohm, G. Sullivan, W.-J. Han, R. Joshi, T. K. Tan, and K. Ugur. Special section on the joint call for proposals on high efficiency video coding (hevc) standardization. *IEEE Transactions on Circuits and Systems for Video Technology*, pages p. 1661 – 1666, 2010. 2
- [6] I. E. G. Richardson. *H.264 and MPEG-4 Video Compression: video coding for next-generation multimedia*. Wiley, 2003. 3, 5, 39, 40, 41, 42, 60
- [7] R. C. Gonzalez and R. C. Woods. *Processamento Digital de Imagens*. Pearson Prentice Hall, 3rd edition, 2010. 5
- [8] Y. Rui, T. S. Huang, and S. Mehrotra. Exploring video structure beyond the shotsn. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pages 237–240. 1998. 5, 6
- [9] N. C. Simões. Detecção de algumas transições abruptas em sequências de imagens. Mestrado, Instituto de Computação, UNICAMP, Campinas, 2004. 5
- [10] T. Lin and H. Zhang. Automatic video scene extraction by shot grouping. *Proceedings of Pattern Recognition*, 4:39–42, 2000. 5
- [11] H. Sundaram and S. Chang. Determining computable scenes in films and their structures using audio-visual memory models. *Proceedings of the eighth ACM international conference on Multimedia*, pages 95–104, 2000. 5

- [12] E. Goubaru and M. Sugisaka. Visual tracking in real-time processing. *SICE-ICASE. International Joint Conference*, pages 5296–5299, 2006. 5
- [13] G. M. Simas, G. P. Fickel, L. Novelo, S. S. C. Botelho, and R. A. de Bem. Using computer vision for 3d probabilistic reconstruction and motion tracking. *Computational Modeling (MCSUL), Third Southern Conference on*, pages 119–124, 2010. 6
- [14] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *Aerospace and Electronic Systems, IEEE Transactions on*, 39(4):1333–1364, 2003. 6
- [15] Z. W. Pylyshyn and R. W. Storm. Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial Vision*, pages 179–197, 1988. 6
- [16] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys (CSUR)*, 38(13):1–45, 2006. 6, 7
- [17] A. Cavallaro, O. Steiger, and T. Ebrahimi. Tracking video objects in cluttered background. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(4):575–584, 2005. 7
- [18] E. Marchand, P. Bouthemy, and F. Chaumette. A 2d-3d model-based approach to real-time visual tracking. *Image and Vision Computing*, 19:941–955, 2001. 8
- [19] Z. Zivkovic and B. Krose. An em-like algorithm for color-histogram-based object tracking. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1:I–798–I–803, 2004. 8
- [20] I. Celasun, A. M. Tekalp, M. H. Gökçetekin, and D. M. Harmanci. 2-d mesh-based video object segmentation and tracking with occlusion resolution. *SPIC*, 16:949–962, 2001. 9
- [21] F. de B. Vidal. *Rastreamento Visual de Objetos utilizando métodos de similaridade de regiões e filtragem estocástica*. Doutorado, Universidade de Brasília, Brasília, 2009. 10, 11, 12
- [22] G. Welch and G. Bishop. An Introduction to the Kalman Filter. Technical Report TR 95-041, University of North Carolina at Chapel Hill, Department of Computer Science, 2006. 11
- [23] E. C. Schopf. Método neuro-estatístico para predição de séries temporais ruidosas. Mestrado, Instituto de Informática. UFRGS, Porto Alegre, 2007. 11
- [24] E. Daeipour and Y. Bar-Shalom. An interacting multiple model approach for target tracking with glint noise. In *Radar Conference, 1994., Record of the 1994 IEEE National*, pages 150–154. IEEE, 1994. 11
- [25] S. McGinnity and G. W. Irwin. Multiple model bootstrap filter for maneuvering target tracking. *Aerospace and Electronic Systems, IEEE Transactions on*, 36(3):1006–1012, 2000. 11

- [26] D. Marimon, Y. Abdeljaoued, B. Palacios, and T. Ebrahimi. Feature point tracking combining the interacting multiple model filter and an efficient assignment algorithms. *Proceedings of the SPIE Electronic Imaging Conf. on Visual Communications and Image Processing (VCIP)*, 6508:650808–650808–8, 2007. 11
- [27] H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *Proceedings of IEEE Transactions on Automatic Control*, 33:780–783, 1988. 11
- [28] C. M. Smith, J. J. Leonard, A. A. Bennett, and C. Shaw. Concurrent mapping and localization for autonomous underwater vehicles. *IEEE Oceans*, 2007. 11
- [29] J. B. Collins and J. K. Uhlmann. Efficient gating in data association with multivariate gaussian distributed states. *Aerospace and Electronic Systems, IEEE Transactions on*, 28(3):909–916, 1992. 12
- [30] S. Challa, M. R. Morelande, D. Musicki, and R. J. Evans. *Fundamentals of object tracking*. Cambridge University Press, 2011. 12, 13
- [31] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Pearson Prentice Hall, 2003. 12
- [32] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Mathematics in Science and Engineering Series. Academic Press, 1988. 12
- [33] O. Songhwi and S. Sastry. A polynomial-time approximation algorithm for joint probabilistic data association. In *American Control Conference, 2005. Proceedings of the 2005*, volume 2, pages 1283–1288. IEEE, 2005. 12
- [34] I. J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10:53–66, 1993. 13
- [35] D. B. Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6):843–854, 1979. 13
- [36] T. Cham and J. M. Rehg. A multiple hypothesis approach to figure tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages –244 Vol. 2. IEEE, 1999. 13
- [37] A. Kaur and M.D. Singh. An overview of pso- based approaches in image segmentation. *International Journal of Engineering and Technology*, 2(8):1349–1357, 2012. 13, 14
- [38] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948. 1995. 14
- [39] D. Floreano and C. Mattiussi. *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*. MIT Press, 2008. 14
- [40] L. B. Booker, D. E. Goldberg, and J. H. Holland. Machine learning: paradigms and methods. chapter Classifier systems and genetic algorithms, pages 235–282. Elsevier North-Holland, Inc., New York, NY, USA, 1990. 14

- [41] S. S. Haykin. *Neural networks: a comprehensive foundation*. Pearson Prentice Hall, 1999. 14
- [42] M. Clerc. *Particle swarm optimization*. ISTE Ltd, 2006. 14, 16
- [43] Dervis Karaboga and Bahriye Akay. A survey: algorithms simulating bee swarm intelligence. *Artif. Intell. Rev.*, 31(1-4):61–85, jun 2009. 14, 15
- [44] R. Poli. Analysis of the publications on the applications of particle swarm optimisation. *J. Artif. Evol. App.*, 2008(1):1–10, jan 2008. 15
- [45] T. Kobayashi, K. Nakagawa, J. Imae, and G. Zhai. Real time object tracking on video image sequence using particle swarm optimization. In COEX, editor, *Proceedings of International Conference on Control, Automation and Systems*. 2007. 19, 20, 22, 23, 24
- [46] R. C. Eberhart, Y. Y. Shi, and J. Kennedy. *Swarm Intelligence*. The Morgan Kaufmann Series in Artificial Intelligence. Elsevier Science, 2001. 19, 20
- [47] R. A. Fisher. The use of multiple measurements in taxonomic problems. In *Annals of Eugenics*, pages 179–188. John Wiley & Sons Ltd., 1936. 25
- [48] J. H. Friedman. Regularized Discriminant Analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989. 25
- [49] J. Lepš and P. Šmilauer. *Multivariate Analysis of Ecological Data Using CANOCO*. Cambridge University Press, 2003. 25
- [50] R. Khattree and D. N. Naik. *Multivariate data reduction and discrimination with SAS software*. SAS Institute Inc., 2000. 25
- [51] J. F. Hair and R. E. Anderson. *Multivariate data analysis*. Prentice Hall Higher Education, 7th edition, 2010. 25, 26, 29, 30, 31, 32
- [52] A. M. Martinez and A. C. Kak. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001. 27, 36
- [53] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition. *Pattern Recogn. Lett.*, 26(2):181–191, 2005. 27
- [54] M. Li and B. Yuan. 2d-lda: A statistical linear discriminant analysis for image matrix. *Pattern Recogn. Lett.*, 26(5):527–532, 2005. 27
- [55] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall, 2007. 28
- [56] E. P. F. Silva. *Advanced Heterogeneous Video Transcoding*. Doutorado, University of London, Queen Mary, School of Electronic Engineering and Computer Science, London, 2012. 33
- [57] I. E. Boulton and L. G. Brown. Factorisation-based segmentation of motions. In *Proceedings of IEEE Workshop on Visual Motion*, pages 179–186. 1991. 39

- [58] B. K. P. Horn and B.G. Schunck. Determining optical flow. *ARTIFICIAL INTELLIGENCE*, 17:185–203, 1981. 40, 41, 49
- [59] R. L. Barbosa, R. B. A. Gallis, J. F. C. Silva, and M. Meneguette Jr. A computação do fluxo óptico em imagens obtidas por um sistema móvel de mapeamento terrestre. *revista Brasileira de Cartografia*, pages p. 72 – 78, 2005. 40
- [60] E. Ota. Tracking red object in a movie using particle filter, 2013. Disponível em: http://www.mathworks.com/matlabcentral/fileexchange/33666-simple-particle-filter-demo/content/PF_Video_EN/particle_filter_by_saved_movie.m. 47