

**RASTREAMENTO VISUAL DE OBJETOS UTILIZANDO
MÉTODOS DE SIMILARIDADE DE REGIÕES E
FILTRAGEM ESTOCÁSTICA**

FLÁVIO DE BARROS VIDAL

**TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**RASTREAMENTO VISUAL DE OBJETOS UTILIZANDO
MÉTODOS DE SIMILARIDADE DE REGIÕES E
FILTRAGEM ESTOCÁSTICA**

FLÁVIO DE BARROS VIDAL

**TESE DE DOUTORADO ACADÊMICO SUBMETIDA AO DEPARTAMENTO DE
ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSI-
DADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A
OBTENÇÃO DO GRAU DE DOUTOR EM ENGENHARIA ELÉTRICA.**

APROVADA POR:

**Prof. V. H. Casanova Alcalde, Ph.D., ENE/UnB
(Orientador)**

**Prof. Dr. Takashi Yoneyama, Ph.D., IEE/ITA
Membro Externo**

**Prof. Gerson Pfitscher, Docteur, CIC/UnB
Membro Externo**

**Prof. José M. S. T. da Motta, Ph.D., ENM/UnB
Membro Externo**

**Prof. João Y. Ishihara, DSc., ENE/UnB
Membro Interno**

BRASÍLIA, 19 DE JUNHO DE 2009.

FICHA CATALOGRÁFICA

DE BARROS VIDAL, FLÁVIO

RASTREAMENTO VISUAL DE OBJETOS UTILIZANDO MÉTODOS DE SIMILARIDADE DE REGIÕES E FILTRAGEM ESTOCÁSTICA [Distrito Federal] 2009.

xvi, 99p., 210 x 297 mm (ENE/FT/UnB, Doutor, Engenharia Elétrica, 2009).

Tese de Doutorado – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Rastreamento Visual

2. Similaridade

3. Filtro de Kalman

4. PSO

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

VIDAL, F. B. (2009). RASTREAMENTO VISUAL DE OBJETOS

UTILIZANDO MÉTODOS DE SIMILARIDADE DE REGIÕES E FILTRAGEM

ESTOCÁSTICA, Tese de Doutorado em Engenharia Elétrica, Publicação

PPGENE.TD-041/09, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 99p.

CESSÃO DE DIREITOS

AUTOR: Flávio de Barros Vidal

TÍTULO: RASTREAMENTO VISUAL DE OBJETOS UTILIZANDO MÉTODOS DE SIMILARIDADE DE REGIÕES E FILTRAGEM ESTOCÁSTICA.

GRAU: Doutor

ANO: 2009

É concedida à Universidade de Brasília permissão para reproduzir cópias desta tese de doutorado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa tese de doutorado pode ser reproduzida sem autorização por escrito do autor.

Flávio de Barros Vidal

Departamento de Eng. Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

*Dedicado a Nerci Vidal Lacerda(in
memorian) e Maria Cezarina de
Barros Vidal.*

AGRADECIMENTOS

Inicialmente a Deus, que me deu saúde, oportunidade para alcançar meus objetivos e pelo melhores presentes da minha vida: Meus sobrinhos Cecilia e Lucas.

Aos meus pais, Nerci Vidal Lacerda (in memorian) e Maria Cezarina de Barros Vidal, pelo apoio e credibilidade. Aos meus irmãos queridos, Hélves e Cínthya, pela dedicação e confiança. Meus cunhados Luiz Carlos e Eliane. Ao grande amor da minha vida, Alda Mieko, pela paciência, compreensão e apoio nos momentos mais difíceis desta jornada. Agradeço aos tios (e/ou "pais") queridos, Márcio e Cecília, pelos bons conselhos e aos seus filhos Ingrid e Exuperry. Tio Jorge Barros pelo incentivo em nunca desistir.

Ao meu orientador, Prof. Victor Hugo Casanova, por ter acreditado e me apoiado neste trabalho. Aos colegas de Pós-Graduação: Carla Aguiar, Bruno Vilhena, Odilon Braz, Bruno Macchiavello, Alaor Arruda, Kenneth Sampaio, Lucas Romano, Marcelino Andrade, George Lauro, Luís Filomeno e Márcio Albertoni, que caminhando juntos me apoiaram e aconselharam.

Aos(as) amigos(as): Gustavo Faria, André Ângelo, Meire Naves, Ana Laura, Davi, Inácio Pinto, Alvaro Pinto, Itamar Guimarães, Amanda, Juliana, Lee Chong, Cláudio Henrique, Keila e Dezaneti, que estiveram presentes nos momentos felizes e difíceis desta caminhada.

RESUMO

RASTREAMENTO VISUAL DE OBJETOS UTILIZANDO MÉTODOS DE SIMILARIDADE DE REGIÕES E FILTRAGEM ESTOCÁSTICA

Autor: Flávio de Barros Vidal

Orientador: Prof. V. H. Casanova Alcalde, Ph.D., ENE/UnB

Programa de Pós-graduação em Engenharia Elétrica

Brasília, Junho de 2009

Inicialmente são apresentadas a importância e as aplicações que envolvem o processo de rastreamento visual de objetos. O problema de rastreamento visual pode ser definido como um processo de estimação em conjunto com técnicas de processamento de imagens. Os principais métodos que realizam rastreamento visual encontrados na literatura são discutidos. O problema tratado aqui consiste em rastreamento automático de um dado objeto que aparece em uma sequência de imagens obtida por um sistema de visão computacional. Neste trabalho propõem-se métodos para a realização desta tarefa baseados na similaridade de regiões, o *window-matching (WM) method*. Este método baseia-se na obtenção de regiões de similaridade em função do padrão da cor cinza de uma janela em torno de um ponto de interesse. Discutem-se também as principais formas de medição da similaridade e a escolha pela função soma do quadrado das diferenças (SSD) é também justificada. Em adição, discutem-se os fatores e parâmetros que afetam o bom desempenho do método tais como: tipo de movimento realizado, oclusões, variação do tamanho da janela, mudanças de iluminação, etc. Desenvolveu-se e implementou-se então um algoritmo de rastreamento (WM) baseado na similaridade de regiões que utiliza a função SSD. O algoritmo foi então aplicado a diversas situações de rastreamento. Observou-se que, para certas aplicações, o algoritmo WM não acompanhava o objeto rastreado. Então, como o rastreamento pode ser tratado como sendo um problema de estimação, introduziu-se um procedimento recursivo para estimação ótima a partir das medidas produzidas pelo algoritmo WM. No processo dinâmico de rastreamento, o vetor de estado a ser estimado consiste dos vetores de posição e velocidade 2D do ponto de interesse, sendo o vetor de medição dado pelos vetores correspondentes fornecidos pelo algoritmo WM. O método leva agora em consideração as características estocásticas do processo de rastreamento (ruídos intrínseco e de medida) e a estimação ótima é realizada pelo filtro de Kalman, que estimará posição e velocidade e as incertezas correspondentes. Um novo algoritmo integrando esta filtragem estocástica (WM+K) foi desenvolvido e implementado. Observou-se que a filtragem estocástica realmente melhora o desempenho do rastreamento. Na procura por aumentar mais ainda a robustez do algoritmo e a sua convergência adicionou-se uma busca heurística nas soluções baseada na otimização seguindo o agrupamento de partículas. Desenvolveu-se assim o algoritmo WM+K+PSO que além de maior robustez produziu trajetórias de rastreamento mais suaves.

ABSTRACT

VISUAL TRACKING OBJECTS USING SIMILARITY TECHNIQUES AND STOCHASTIC FILTERING

Author: Flávio de Barros Vidal

Supervisor: Prof. V. H. Casanova Alcalde, Ph.D., ENE/UnB

Programa de Pós-graduação em Engenharia Elétrica

Initially the importance and applications on which object visual tracking is involved are presented. Visual tracking problem can be stated as an estimation process acting together with digital image processing techniques. The different methods found in the literature for solving this tracking problem are discussed. The problem being dealt here consists of automatic tracking of a given object appearing in a sequence of images captured by a computer vision system. This work proposes methods to perform this task that are based on the *window-matching (WM) techniques*. These techniques are based on obtaining similar regions in terms of the gray level pattern of a window around a point of interest. The option for these techniques are justified and the main hypotheses are discussed. The different ways of measuring similarity are also discussed and the choice of the sum of square differences (SSD) as a similarity cost function is also justified. A discussion follows of situations that affect the tracking results, as type of motion, occlusions, window size variations, illumination changes, etc. An object tracking algorithm (WM) based on regions of similarity as measured by the SSD cost function is developed and implemented. The algorithm is then applied for tracking objects in different situations. It was observed, for certain applications, that the WM algorithm failed to track the object. Then, as tracking can be considered an estimation problem, a recursive procedure for optimal estimation from measurements generated by the WM algorithm. In the tracking dynamical process the state vector consist of the 2D position and velocity coordinates of the point of interest, being the measurement vector the corresponding output from the WM algorithm. The new method now takes into account the stochastic properties of the tracking process (intrinsic and measurement noise) and the optimal estimation is performed by the Kalman filter, being the output estimates of the position and velocity and the corresponding uncertainties. A new algorithm integrating this stochastic filtering (WM+K) is developed and implemented. Indeed the stochastic filtering improves the tracking performance and succeeds where the WM fails. Further procedures to increase the robustness and convergence of the tracking algorithm were pursued. Introducing a heuristic search based on Particle Swarm Optimization allowed to obtain smooth tracking trajectories.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	UMA BREVE REVISÃO BIBLIOGRÁFICA.....	1
1.2	ORGANIZAÇÃO DA TESE	3
2	RASTREAMENTO VISUAL	5
2.1	CONCEITOS BÁSICOS SOBRE RASTREAMENTO VISUAL	5
2.2	MÉTODOS DETERMINÍSTICOS	6
2.2.1	RASTREAMENTO POR SEGMENTAÇÃO	6
2.2.2	RASTREAMENTO DE CONTORNOS E BORDAS	9
2.2.3	RASTREAMENTO DE UMA REGIÃO DE INTERESSE	9
2.3	MÉTODOS ESTOCÁSTICOS	12
2.3.1	RASTREAMENTO POR ESTIMAÇÃO DE ESTADOS	12
2.3.2	RASTREAMENTO POR ASSOCIAÇÃO DE DADOS.....	14
3	RASTREAMENTO BASEADO NA SIMILARIDADE DE REGIÕES	17
3.1	MÉTODOS BASEADOS EM SIMILARIDADE DE REGIÕES	17
3.2	ASPECTOS NA IMPLEMENTAÇÃO DOS MÉTODOS DE SIMILARIDADE.....	18
3.2.1	CARACTERÍSTICAS DO PADRÃO DA REGIÃO DE INTERESSE	18
3.2.2	ESCOLHA DA MEDIDA DE SIMILARIDADE	19
3.2.3	CONFIABILIDADE DAS MEDIDAS DE SIMILARIDADE.....	20
3.2.4	OCLUSÃO DE OBJETOS.....	21
3.3	UM MÉTODO DE RASTREAMENTO BASEADO EM SIMILARIDADES DE RE- GIÕES	22
3.3.1	A FUNÇÃO SOMATÓRIA DAS DIFERENÇAS QUADRÁTICAS (<i>Sum of Square Differences - SSD</i>)	22
3.3.2	VARIAÇÕES DA FUNÇÃO <i>SSD</i> - DOMÍNIO DA FREQUÊNCIA	22
3.4	DESENVOLVIMENTO DE UM ALGORITMO DE RASTREAMENTO BASEADO NA CORRESPONDÊNCIA ENTRE JANELAS.....	25
3.4.1	DESCRIÇÃO DO ALGORITMO WM.....	26
3.5	APLICAÇÕES DO ALGORITMO DE RASTREAMENTO WM.....	28
3.5.1	RASTREAMENTO DA BOLA NA SEQÜÊNCIA DE JOGO DE TÊNIS DE MESA	28
3.5.2	RASTREAMENTO DE VEÍCULO EM SEQÜÊNCIA DE TRÂNSITO.....	28
3.6	COMENTÁRIOS SOBRE O ALGORITMO WM.....	31
4	FILTRAGEM ESTOCÁSTICA NO RASTREAMENTO VISUAL	33
4.1	FILTRO DE KALMAN	33

4.2	O FILTRO DE KALMAN DISCRETO	34
4.2.1	FORMULAÇÃO DO PROBLEMA	34
4.2.2	O ALGORITMO DE FILTRAGEM DE KALMAN	36
4.3	O FILTRO DE KALMAN APLICADO AO RASTREAMENTO VISUAL	37
4.3.1	RASTREAMENTO VISUAL POR CORRESPONDÊNCIA DE JANELAS COM FILTRO DE KALMAN - ALGORITMO WM+K	39
4.4	APLICAÇÕES DO ALGORITMO DE RASTREAMENTO WM+K	40
4.4.1	RASTREAMENTO DE INDIVÍDUOS EM SEQUÊNCIAS DE CIRCULA- ÇÃO DE PESSOAS EM LOCAIS PÚBLICOS	40
4.4.2	RASTREAMENTO DE GARRAFA FLUTUANDO NO MAR	43
5	OTIMIZAÇÃO SEGUNDO AGRUPAMENTO DE PARTÍCULAS APLI- CADA AO RASTREAMENTO VISUAL	49
5.1	UM BREVE HISTÓRICO	49
5.2	O MÉTODO PSO CLÁSSICO	50
5.3	MÉTODO PSO COM A RESTRIÇÃO DE LIPSCHITZ	51
5.4	MÉTODO DE RASTREAMENTO POR CORRESPONDÊNCIA DE JANELAS COM FILTRAGEM ESTOCÁSTICA E BUSCA PSO	52
5.4.1	DESENVOLVIMENTO DE UM ALGORITMO DE RASTREAMENTO POR CORRESPONDÊNCIA DE JANELAS COM FILTRAGEM DE KALMAN E OTIMIZAÇÃO SEGUNDO AGRUPAMENTO DE PARTÍCULAS - ALGO- RITMO WM+K+PSO	52
5.5	APLICAÇÕES DO ALGORITMO WM+K+PSO NO RASTREAMENTO VISUAL ..	54
5.5.1	RASTREAMENTO DE INDIVÍDUOS EM SEQUÊNCIA DE CIRCULAÇÃO DE PESSOAS EM LOCAIS PÚBLICOS	54
5.5.2	RASTREAMENTO DE GARRAFA FLUTUANDO NO MAR	55
5.5.3	RASTREAMENTO DE FACE HUMANA COM OCORRÊNCIA DE OCLU- SÕES PARCIAIS	56
5.5.4	RASTREAMENTO DE UM OBJETO MOVIMENTADO DE FORMA ALE- ATÓRIA POR UM INDIVÍDUO	58
5.6	TABELA COMPARATIVA DO CUSTO COMPUTACIONAL DOS ALGORITMOS DESENVOLVIDOS	60
6	CONCLUSÕES	62
	REFERÊNCIAS BIBLIOGRÁFICAS	65
	ANEXOS	73
A	DIAGRAMA DE BLOCOS DOS ALGORITMOS DESENVOLVIDOS ...	74
A.1	ALGORITMO PARA CÁLCULO DO CENTRÓIDE DE IMAGENS SEGMENTADAS	74

B	SEQÜÊNCIAS DE IMAGENS	75
B.1	SEQUÊNCIA TÊNIS DE MESA	75
B.2	SEQUÊNCIA TRÂNSITO URBANO	77
B.3	SEQUÊNCIA GARRAFA FLUTUANDO NO MAR	79
B.4	SEQUÊNCIA ENCONTRO NO SHOPPING	82
B.5	SEQUÊNCIA ENCONTRO NO SHOPPING COM OCLUSÃO PARCIAL	86
B.6	SEQUÊNCIA DE FACE HUMANA COM OCLUSÕES PARCIAIS	90
B.7	SEQUÊNCIA OBJETO COM MOVIMENTOS DIVERSOS	92
C	VARIAÇÕES DO FILTRO DE KALMAN	94
C.1	FILTRO DE KALMAN ESTENDIDO	94
C.1.1	EQUAÇÕES DO FILTRO DE KALMAN ESTENDIDO	94
C.2	FILTRO DE KALMAN UNSCENTED	97

LISTA DE FIGURAS

2.1	Fluxograma do Rastreamento por Segmentação de Movimento.	7
2.2	Fluxograma Etapa de Pós-processamento do Rastreamento por Segmentação de Movimento.	7
2.3	Rastreamento por Segmentação de Movimento utilizando Fluxo Óptico Diferencial.	8
2.4	Fluxograma do Rastreamento por Segmentação de Cores.	8
2.5	Rastreamento por Segmentação de Cores de imagens térmicas.	8
2.6	Fluxograma do Rastreamento por Bordas utilizando Energia da Imagem.	10
2.7	Rastreamento por extração de bordas da imagem. Em (a) a imagem original, (b) extração dos limites utilizando a metodologia de Canny e em (c) o resultado do rastreamento das bordas.	10
2.8	Fluxograma do Rastreamento por região de interesse na forma Padrão.	11
2.9	Em (a) a imagem no instante k com a região de interesse formada em torno do ponto marcado em verde e em vermelho em (b) a imagem no instante $k + I$ o pixel correspondente. (c) é a matriz de similaridade da função de custo para a região de interesse em (a).	12
2.10	Diagrama da Metodologia de rastreamento por estimativa de estados.	13
2.11	Fluxograma da Metodologia de rastreamento por filtro IMM utilizando dois modelos para estimação dos estados.	14
3.1	Fluxograma do Algoritmo Window-matching.	25
3.2	Resultado do rastreamento realizado pelo algoritmo WM utilizando a função de Custo SSD para seqüência do Jogo de Tênis de Mesa.	29
3.3	Resultado do rastreamento realizado pelo algoritmo WM utilizando a função de Custo SSD para seqüência do jogo de Tênis de Mesa nas direções horizontal (a) e vertical (b). Legenda: \square WM e em \triangle Real	29
3.4	Erro de trajetória obtido pelo algoritmo Window-matching em comparação com a trajetória real realizada pelo objeto para a seqüência do jogo de Tênis de Mesa. Em (a) com relação a direção horizontal e em (b) na direção vertical.	30
3.5	Resultado do rastreamento realizado pelo algoritmo WM utilizando a função de Custo SSD para seqüência de Trânsito	30
3.6	Resultado do rastreamento realizado pelo algoritmo WM utilizando a função de Custo SSD para seqüência de Trânsito nas direções horizontal (a) e vertical (b). Legenda: \square WM e em \triangle Real	31
3.7	Erro de trajetória obtido pelo algoritmo Window-matching em comparação com a trajetória real realizada pelo objeto. Em (a) com relação a direção horizontal e em (b) na direção vertical.	31

4.1	Dinâmica da execução do Algoritmo do Filtro de Kalman.....	36
4.2	Fluxograma do Algoritmo Window-matching com etapa de Filtragem de Kalman (WM+K)	39
4.3	Resultado do rastreamento realizado pelo algoritmo Window-matching utilizando a função de Custo SSD para seqüência Encontro no Shopping.	41
4.4	Resultado do rastreamento realizado pelo algoritmo WM+K para seqüência Encontro no Shopping.	42
4.5	Posições horizontal(a) e vertical(b) descritas pelo rastreamento WM+K para seqüência Encontro no Shopping. Legenda: \square WM e em \triangle WM+K.....	43
4.6	Resultado do rastreamento realizado pelo algoritmo WM para seqüência Caminhada no Shopping.....	44
4.7	Resultado do rastreamento realizado pelo algoritmo WM+K para seqüência Caminhada no Shopping.....	45
4.8	Posições horizontal(a) e vertical(b) descritas pelo rastreamento WM+K para seqüência Caminhada no Shopping. Legenda: \square WM e em \triangle WM+K	45
4.9	Resultado do rastreamento realizado pelo algoritmo WM para seqüência objeto flutuando no mar.	46
4.10	Resultado do rastreamento realizado pelo algoritmo WM+K para seqüência objeto flutuando no mar.	47
4.11	Posições horizontal(a) e vertical(b) descritas pelo rastreamento WM+K para seqüência objeto flutuando no mar. Legenda: \square WM e em \triangle WM+K	47
5.1	Algoritmo PSO Clássico com restrição de Lipschitz	53
5.2	Metodologia de Acoplamento dos métodos PSO e Window-Matching com Filtro de Kalman.....	53
5.3	Resultado do rastreamento realizado pelo algoritmo WM+K+PSO para seqüência Caminhada no Shopping.	55
5.4	Posições horizontal(a) e vertical(b) descritas pelo rastreamento WM, WM+K e WM+K+PSO para seqüência Caminhada no Shopping. Legenda: \square WM, em \triangle WM+K e em $- * -$ WM+K+PSO.....	56
5.5	Resultado do rastreamento realizado pelo algoritmo WM+K+PSO para seqüência objeto flutuando no mar.	57
5.6	Posições horizontal(a) e vertical(b) descritas pelo rastreamento WM, WM+K e WM+K+PSO para seqüência objeto flutuando no mar. Legenda: \square WM, em \triangle WM+K e em $- * -$ WM+K+PSO.....	57
5.7	Resultado do rastreamento realizado pelo algoritmo WM+K+PSO para seqüência Face humana com oclusões parciais.....	58
5.8	Posições horizontal(a) e vertical(b) descritas pelo rastreamento WM, WM+K e WM+K+PSO para seqüência rastreamento de face humana com oclusões parciais. Legenda: \square WM, em \triangle WM+K e em $- * -$ WM+K+PSO.....	59

5.9	Resultado do rastreamento realizado pelo algoritmo WM+K+PSO para seqüência objeto com movimentos aleatórios. Legenda: Marcação em vermelho - WM, em amarelo - WM+K e verde - WM+K+PSO	59
5.10	Posições horizontal(a) e vertical(b) descritas pelo rastreamento WM, WM+K e WM+K+PSO para seqüência rastreamento objeto com movimentos aleatórios. Legenda: \square WM, em \triangle WM+K e em $- * -$ WM+K+PSO.....	60
5.11	Custo computacional em operações com ponto flutuante dos algoritmos desenvolvidos para as seqüências utilizadas nos testes.	61
A.1	Diagrama de Blocos do Cálculo do Centróide de Imagens Segmentadas.....	74

LISTA DE TABELAS

3.1	Principais funções de Custo utilizadas na medição da similaridade.....	19
-----	--	----

LISTA DE SÍMBOLOS

$J(\vec{p})$	Função de Custo no ponto p	
$\vec{p}(x, y)$	Vetor bidimensional de posição espacial na imagem	[pixels]
W	Largura da imagem	[pixels]
h	Altura da imagem	[pixels]
x	Eixo horizontal na imagem	
y	Eixo vertical na imagem	
χ_k^x	Vetor com <i>sigma points</i> do Filtro de Kalman Unscented em relação ao vetor de estado x no instante k	
i	Incremento inteiro unitário na direção horizontal	
j	Incremento inteiro unitário na direção vertical	
\mathbb{R}^n	Conjunto dos números reais dimensão n	
w_k	Ruído de processo no instante de tempo k	
v_k	Ruído de medição no instante de tempo k	
Q	Matriz de covariância do ruído de processo	
R	Matriz de covariância do ruído de medição	
$E(\mathbf{x}_k)$	Valor Esperado do vetor de estados x no instante de tempo k	
$\mathbf{W}^{(m)}$	Matriz de pesos simetricamente distribuídos com dimensão m	
n	Índice indicativo de número do frame/imagem em análise	
\rightarrow	Notação para vetores	
\bar{x}	Notação para média de um número ou vetor x	
g^*	Complexo conjugado da função g	
<i>ROI</i>	Região de Interesse na Imagem	
<i>IMM</i>	Interação de Múltiplos Modelos	
<i>NN</i>	Vizinhança-próxima	
<i>JPDA</i>	Associação Probabilística Conjunta de Dados	
<i>MHT</i>	Rastreamento por Múltiplas Hipóteses	
<i>SAD</i>	Soma das Diferenças Absolutas	
<i>SSD</i>	Soma do Quadrado das Diferenças	
<i>ZSAD</i>	Soma das Diferenças Absolutas Globais	
<i>ZSSD</i>	Soma do Quadrado das Diferenças Globais	
<i>LSAD</i>	Soma das Diferenças Absolutas Locais	
<i>LSSD</i>	Soma do Quadrado das Diferenças Locais	

<i>CC</i>	Correlação-cruzada
<i>NCC</i>	Correlação-cruzada Normalizada
<i>ZNCC</i>	Correlação-cruzada Normalizada com média zero
<i>PSO</i>	Particle Swarm Optimization
<i>DFT</i>	Transformada Discreta de Fourier
<i>JPEG</i>	Joint Photographic Experts Group
<i>PNG</i>	Portable Network Graphics
<i>EKF</i>	Filtro de Kalman Estendido
<i>UKF</i>	Filtro de Kalman Unscented
<i>AUKF</i>	Filtro de Kalman Unscented Aditivo
<i>RBUKF</i>	Filtro de Kalman Unscented Rao-Blackwillised
<i>RBAUKF</i>	Filtro de Kalman Unscented Rao-Blackwillised Aditivo

1 INTRODUÇÃO

O rastreamento visual de um objeto em movimento por um observador humano consiste no acompanhamento do movimento do objeto pelo sistema de visão do observador. A visão vai capturando as imagens do objeto e o cérebro as processa de modo que seja possível seguir a trajetória descrita. As diversas aplicações deste processo, bem como a necessidade da sua automação, levaram à tentativa de implementação do rastreamento por uma máquina dotada de um sistema de visão artificial para captura de imagens e de um sistema de processamento digital destas imagens para acompanhar a trajetória sem perder de vista o objeto.

O processo de rastreamento visual consiste em encontrar uma região de interesse nos sucessivos quadros de uma seqüência de imagens, capturadas em um curto intervalo de tempo. Este processo tem se tornado um recurso importante em várias aplicações como: Controle de sistemas baseado em visão [1], [2], robótica industrial [3], biomedicina [4], vigilância [5], rastreamento de alvos aéreos [6], monitoramento e controle de aeronaves e veículos terrestres[7], controle de linha de montagem automotiva [8] e rastreamento de faces humanas[9], entre diversas outras aplicações de pesquisa e desenvolvimento.

Neste trabalho é proposto um algoritmo de rastreamento visual de objetos, que combina técnica *window-matching* com teoria de estimação ótima linear estocástica utilizando a filtragem de Kalman [10]. O algoritmo *window-matching* apresentado foi modificado para ser utilizado em situações que envolvem funções de custo, e posteriormente acoplado ao processo de filtragem estocástica utilizando o Filtro de Kalman. O principal objetivo deste trabalho foi desenvolver uma estrutura para algoritmos de rastreamento, ainda não vislumbrando sua forma final e implementação eficiente, justificando a utilização da plataforma de desenvolvimento computacional Matlab[®]. A consequência natural, e necessária, foram melhorias tanto na robustez, quanto nos resultados de trajetórias dos objetos de interesse em se rastrear. Com o intuito de aprimorar as técnicas de rastreamento já desenvolvidas, e ainda utilizar de recursos que a partir do aumento da capacidade de processamento atuais, optou-se no acoplamento do algoritmo com filtragem estocástica à técnicas de Otimização de Trajetórias por Enxame de Partículas (i.e. *Particle Swarm Optimization*). Estas, por sua vez, tornaram o algoritmo modificado com maior robustez e ainda melhorando a descrição de trajetórias dos alvos a serem rastreados.

1.1 UMA BREVE REVISÃO BIBLIOGRÁFICA

Um dos primeiros trabalhos que envolvem técnicas de rastreamento visuais podem ser encontrados em [11]. Sua ênfase não estava somente no desenvolvimento de técnicas para o rastreamento (neste caso vídeo com baixa taxa de compressão) mas no desenvolvimento de

novas tecnologias para aplicações em tempo real, que envolviam o rastreamento visual para área militar. As primeiras aplicações tiveram como objetivo o rastreamento de objetos que não possuíam mudanças de aparências (forma, escala e rotações por exemplo) e no plano de fundo (*background*) do ambiente durante o rastreamento. Devido às restrições de baixa capacidade computacional da época, este processo de rastreamento só era possível com a utilização de equipamentos dedicados e de alto desempenho. Somente dessa forma as tarefas relativas ao rastreamento poderiam ser realizadas em tempo real.

No trabalho desenvolvido em [12], faz-se o uso do rastreamento visual por câmeras com fibra óptica conectadas no *end-effector* (também chamado de ponta-de-ferramenta) de manipuladores robóticos. Este sistema é caracterizado pelo uso de técnicas de extração de campo de movimento em imagens, adaptadas para o rastreamento visual de objetos em imagens digitais. Desenvolveu-se uma metodologia na qual realiza os cálculos dos parâmetros cinemáticos do manipulador por meio do rastreamento visual alcançar o objeto a ser manipulado. Aplicou-se este algoritmo no desenvolvido de uma célula de montagem, na qual o robô manipulador era responsável em coletar objetos em uma esteira em movimento. Todo o posicionamento dos objetos eram obtidos por meio do algoritmo de rastreamento visual no plano da imagem coletadas por câmeras calibradas sobre o manipulador.

Em [13], foi apresentado os primeiros trabalhos envolvendo rastreamento visual por técnicas de fluxo óptico¹ baseado na similaridade de regiões na imagem. Este método tem como estrutura básica a utilização de funções de custo em aplicações de rastreamento bidimensional em imagens. Discute-se a idéia que a extensão para casos que envolvam rastreamento tridimensional é uma evolução natural, pois o algoritmo desenvolvido interpreta este tipo de situação sem a necessidade de grandes ajustes. Por se tratar da utilização de adaptações de técnicas de fluxo óptico, além da elevada carga computacional utilizada na realização método, não foi feito o uso de nenhuma seqüência de imagens reais (e/ou imagens que represente situações diversas que poderiam ser encontradas durante um processo real de rastreamento).

Ao se abordar a combinação rastreamento visual e técnicas de controle modernas, têm-se o trabalho desenvolvido em [15]. Neste trabalho uma estrutura de rastreamento visual utilizando o modelo do sistema formado pela transformação *objeto-câmera* que interagem no plano da imagem com aplicações para robôs móveis. A estratégia de controle desenvolvida envolve métodos de estabilidade assintótica com o fechamento da malha de controle por meio das informações fornecidas pelo sistema de rastreamento visual. O método de rastreamento utilizado para o módulo de visão foi proposto inicialmente em [16]. Na metodologia apresentada em [16] desenvolveu-se um algoritmo para rastreamento de pessoas, por modelos baseados em objetos estáticos no plano da imagem. Esta metodologia insere na malha de controle do sistema o mapa de velocidade extraído da imagem, em função da posição do objeto rastreado. Dessa forma, todas as informações são processadas de forma a satis-

¹Maiores informações a respeito do desenvolvimento de técnicas de extração de movimento baseados em imagens digitais podem ser encontradas em [14].

fazer teoricamente o critério de estabilidade direta de *Lyapunov*. O sistema desenvolvido não atende aos requisitos de robustez para diversas situações reais, sendo que foi somente aplicado em simulações computacionais para o rastreamento de objetos em movimento.

Tratando-se de métodos de rastreamento visual, nos quais utilizam filtro estocásticos para realizar a tarefa de rastreamento, têm-se o trabalho apresentado em [17]. A proposta inicial deste trabalho é o desenvolvimento de um algoritmo para rastreamento visual que envolvam ambientes com diversos objetos, o que eleva a necessidade do desenvolvimento de algoritmos robustos. Para isso, fez-se o uso do filtro probabilístico para associação de dados. Definiu-se a estimação das posições baseadas nas variáveis de medições. O processo de medição é feito pelas informações fornecidas com a transformada *Hough*[18] das regiões de interesse circulares em torno do objeto a ser rastreado. Em seguida, ao ser aplicado para situações de rastreamento para múltiplos objetos, fez-se o uso de um filtro de associação conjunta de dados para realizar a interação entre as diversas medições provenientes do algoritmo. Ainda na linha de rastreamento visual com etapas de filtragem estocástica, em [19] utiliza também a transformada *Hough*, entretanto ao invés de alimentar um filtro probabilístico para realizar a associação de dados, fez-se o uso do filtro de rastreamento por múltiplas hipóteses, seguindo de forma semelhante a utilizada no trabalho de [20]. Neste caso as imagens utilizadas para teste são imagens sintéticas com elevado contraste em relação ao fundo da imagem.

Algoritmos que combinam processamento de imagens digitais e técnicas de controle servo-visual, estão sendo aplicados na solução de problemas complexos tais como rastreamento de objetos a partir de uma seqüência de imagens [21]. O processo de rastreamento também pode ser considerado como um processo de estimação agindo em conjunto com processamento de imagens digitais. Para o processo de estimação pode-se utilizar procedimentos de filtragem estocástica, como o Filtro de Kalman [22] e o Filtro de Partículas [4]. Um algoritmo para rastreamento visual descrito em [23] combina um rastreador de média-móvel com uma modalidade de algoritmo de similaridade de regiões (*window-matching*) para minimizar o efeito de *drift* durante a ocorrência de uma oclusão parcial. Outro algoritmo proposto em [24] utiliza o filtro de partículas para o rastreamento de objetos baseado em múltiplas sugestões (i. e. *cues*) em conjunto com parâmetros adaptativos e seu desempenho é avaliado com relação à imagens sintéticas e reais, comparadas com o rastreador de média-móvel. Estes processos de estimação podem ser aplicados em processos de controle servo-visual em associação com algoritmos do tipo *window-matching* obtendo assim melhores resultados [25].

1.2 ORGANIZAÇÃO DA TESE

A organização da Tese foi definida como apresentado abaixo.

No capítulo 2 são apresentados os principais conceitos e metodologias que abrangem as

técnicas de rastreamento visual encontradas na vasta literatura. Realizou-se um processo de classificação dos tipos de rastreamento visual, classificando cada metodologia que utilizaram estrutura de desenvolvimento baseada em métodos determinísticos ou estocásticos. Permitindo melhor entendimento e descrevendo as contribuições realizadas durante todo o processo de pesquisa no assunto.

No capítulo 3 são discutidos as principais técnicas de rastreamento por similaridade entre regiões de interesse, suas estruturas fundamentais e os algoritmos desenvolvidos, bem como resultados do emprego deste no processo de rastreamento. Também é apresentado neste capítulo um algoritmo desenvolvido a partir das técnicas pesquisadas no qual ficou conhecido como *window-matching*.

No capítulo 4 o Filtro de Kalman e suas principais características são apresentadas, juntamente com o processo acoplamento com o algoritmo *window-matching* desenvolvido e os resultados dos testes realizados, gerando o algoritmo definido WM+K. A modelagem da estrutura desenvolvida foi investigada no qual se empregou o processo de filtragem de Kalman para aumentar a robustez e transformar o algoritmo *window-matching* aplicável para diversas situações que podem ser encontradas no rastreamento visual de objetos.

Continuando o que foi apresentado no capítulo 4, têm-se no capítulo 5 a apresentação da técnica de otimização por enxame de Partículas - PSO, bem como suas principais formulações e sua integração ao algoritmo desenvolvido. Esta integração serviu para melhorar o desempenho do algoritmo WM+K, e também, para realizar melhorias na trajetória descritas pelo rastreamento.

Para a finalização deste trabalho, apresenta-se no capítulo 6 as principais conclusões retiradas durante todo o processo de desenvolvimento da pesquisa realizada, bem como propostas para trabalhos futuros a serem desenvolvidos.

2 RASTREAMENTO VISUAL

Neste capítulo descrevem-se as técnicas mais relevantes de rastreamento visual.

2.1 CONCEITOS BÁSICOS SOBRE RASTREAMENTO VISUAL

Nos anos recentes, houve um grande aumento de situações e aplicações que utilizam o rastreamento visual em imagens digitais. O surgimento de novas metodologias de rastreamento visual para um único ou vários objetos (*Multitarget Tracking*), despertou grande interesse em pesquisas e aplicações. Elas envolvem diversas áreas como biomedicina [2], automação de processos [3], indústria bélica [6], sistemas de monitoramento e controle de tráfego de veículos e aeronaves [7], sistemas de vigilância [5], entre outras.

Em [26], define-se como rastreamento visual o procedimento de localizar uma ou mais características desejadas em uma seqüência de imagens. Uma forma de realização do rastreamento visual pode ser definida por métodos de detecção de movimento utilizando derivadas espaciais e temporais das imagens que compõem a seqüência em análise [27]. Outras variações também podem ser encontradas em [28], que define como rastreamento visual o quanto regiões na imagem são similares, identificando o movimento de regiões como rastreamento.

Para identificar os elementos que compõem sistemas de rastreamento visual, seque-se aqui a terminologia utilizada neste trabalho:

- *Estado do Alvo* - vetor de pequena dimensão que armazena as informações passadas do alvo, o suficiente para ser utilizado em operações de estimações futuras.
- *Medição* - observação ruidosa correlacionada ao estado do alvo.
- *Rastro* - conjunto de medidas no tempo associadas a algum alvo.
- *Sensor* - unidade responsável pela aquisição de informações do ambiente.
- *Quadro* - captura de região do ambiente obtido pelo sensor em um instante de tempo, chamado de *intervalo de amostragem*.
- *Mudança de Posição (manobra)* - mudanças não estimadas da posição do alvo.
- *Região de Interesse (ROI)* - região delimitada que contém o alvo a ser rastreado.

Neste trabalho, os *alvos* correspondem aos pontos centrais das regiões de interesse e suas características na imagem. O *sensor* é equivalente à câmera, independente de seus parâmetros construtivos. Para descrever as metodologias existentes para o rastreamento visual,

poderia optar-se por classificá-las de acordo com suas principais características e estruturas de concepção seguindo o proposto em [29]. Com o passar dos anos e o crescimento da capacidade computacional, esta classificação não pode mais ser utilizada, pois não agrega os métodos que envolvem características estocásticas. Desta forma, optou-se em dividir os principais métodos em *determinísticos* e *estocásticos*. Deve-se ressaltar também, que esta classificação é feita de forma a melhor correlacionar as metodologias de rastreamento visual existentes à estrutura deste trabalho.

2.2 MÉTODOS DETERMINÍSTICOS

Com relação aos métodos determinísticos, optou-se em subdividí-los em três grupos. Estes grupos são: *Rastreamento por Segmentação*, *Rastreamento por Contornos e Bordas* e *Rastreamento de uma Região de Interesse*.

2.2.1 Rastreamento por Segmentação

O processo de segmentação de imagens envolve técnicas de processamento que permitem destacar determinadas regiões do restante da imagem, utilizando critérios pré-definidos [18]. Tais critérios podem ser baseados em informações provenientes da imagem como cor, intensidade e movimento dos pixels que representam o alvo a ser rastreado. Determinada a posição da imagem correspondente ao alvo, sua posição pode ser obtida através do centróide da região formada.

Como parte do trabalho de pesquisa desenvolveram-se dois algoritmos de rastreamento por segmentação. O primeiro algoritmo [27] realiza o rastreamento pela segmentação da intensidade de movimento dos pixels. Ele é baseado na determinação do fluxo óptico diferencial de Horn & Schunck [30] que representa o movimento dos pixels. As informações deste movimento são obtidas a partir de suas respectivas derivadas, determinando-se então seus respectivos campos de movimento. Variações desta metodologia que incluem funções diversas para extração do movimento em seqüências de imagens foram introduzidas por [31] e [32]. As figuras 2.1 e 2.2 mostram o fluxograma do algoritmo desenvolvido, sua aplicação ao rastreio de uma pessoa é mostrada na figura 2.3.

O segundo algoritmo desenvolvido [29] realiza o rastreamento pela segmentação de cores. A figura 2.4 mostra o fluxograma do algoritmo e sua aplicação ao rastreamento é mostrada na figura 2.5. O algoritmo demanda carga computacional reduzida e por isso foi aplicado a uma competição de futebol de robôs, na qual o rastreamento deve ser realizado em tempo real.

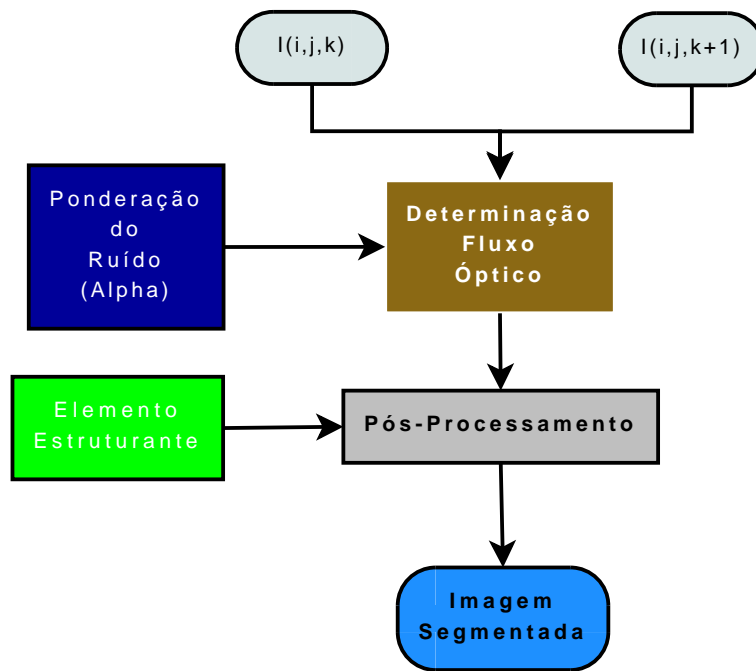


Figura 2.1: Fluxograma do Rastreamento por Segmentação de Movimento.

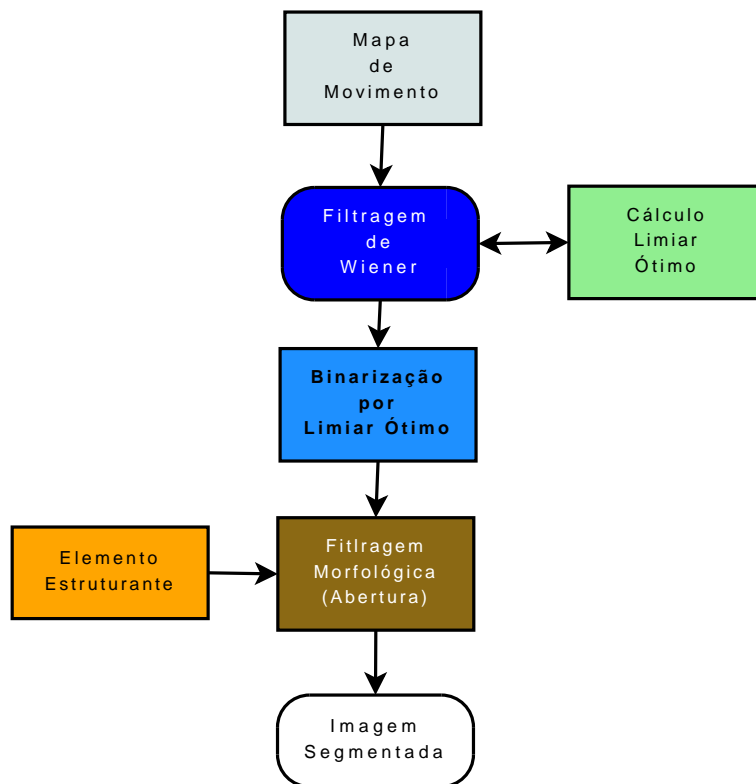
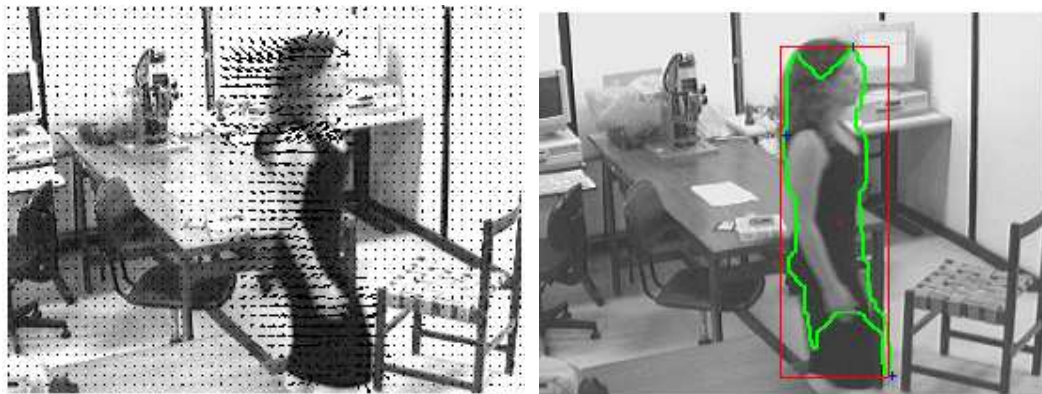


Figura 2.2: Fluxograma Etapa de Pós-processamento do Rastreamento por Segmentação de Movimento.



(a)

(b)

Figura 2.3: Rastreamento por Segmentação de Movimento utilizando Fluxo Óptico Diferencial.

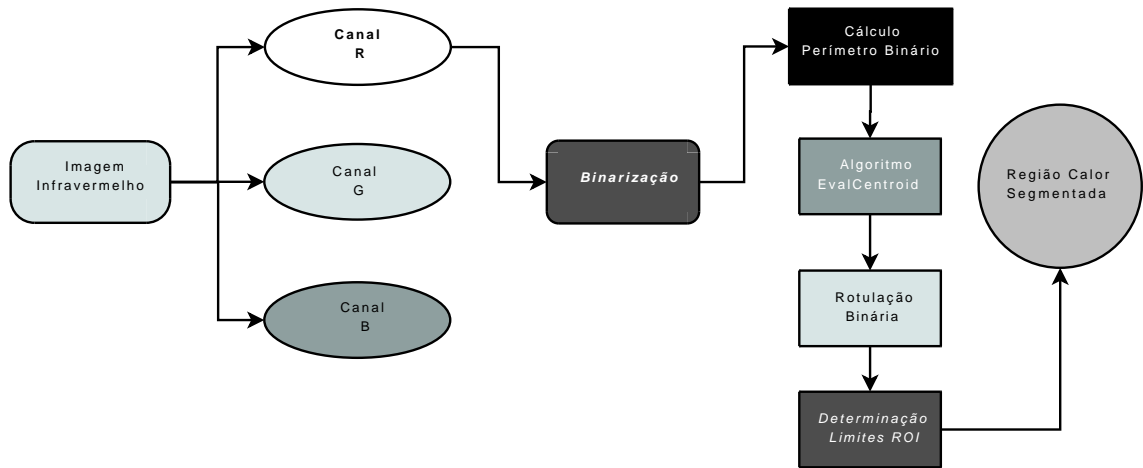
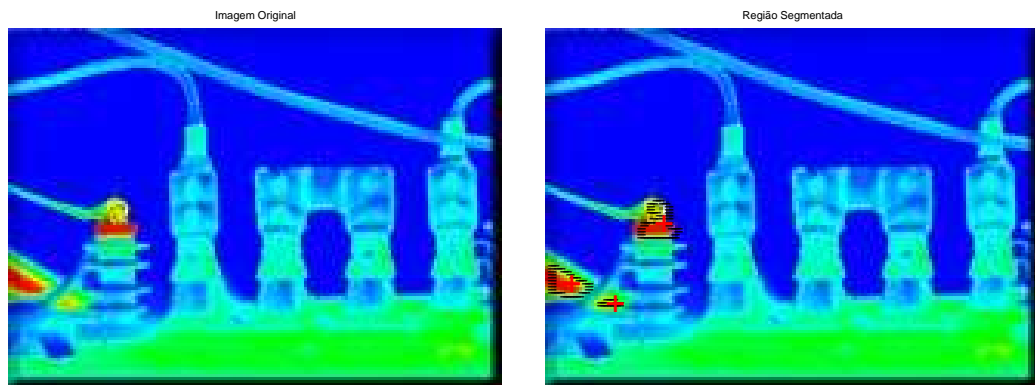


Figura 2.4: Fluxograma do Rastreamento por Segmentação de Cores.



(a)

(b)

Figura 2.5: Rastreamento por Segmentação de Cores de imagens térmicas.

2.2.2 Rastreamento de Contornos e Bordas

Os contornos e bordas em uma imagem estão associados às características geométricas dos alvos a serem rastreados [29]. Diversos elementos formadores das características geométricas do objeto de interesse podem ser utilizados como parâmetros para o rastreamento. Elementos componentes da imagem são fortes influenciadores no processo de obtenção de informações de contorno e bordas. Tais elementos, como a iluminação, fundo da imagem (*background*) e oclusões por exemplo, podem comprometer o processo de extração de contornos, e conseqüentemente, o rastreamento.

Uma forma de se identificar elementos geométricos é pela utilização de funções paramétricas. A principal aplicação de tais funções é na obtenção de parâmetros que indiquem a propriedade do objeto a ser rastreado (posições, inclinações, raios,...). Em [9] utiliza-se o método de extração de bordas de Canny [33] para realizar o rastreamento de faces humanas, onde a metodologia é proposta para imagens em diversas situações complexas. A partir das bordas extraídas, um mecanismo envolvendo modelo visual-neural realiza a extração das funções paramétricas utilizadas no rastreamento. Normalmente, utiliza-se a *Transformada Hough* [18] para extrair funções paramétricas de retas e circunferências em imagens e em seguida realizar o rastreamento de pontos de interesse. Em [34] utiliza-se a *Transformada Hough* para fazer a identificação de uma bola, em conjunto com redes neurais artificiais. Na literatura, também podem-se encontrar métodos de rastreamento por contornos que envolvam a energia dos pixels da região de borda, como o método proposto por Kitchen e Rosenfeld [35]. Como parte do trabalho de pesquisa desenvolveu-se um algoritmo que executa o rastreamento pela extração de bordas segundo a metodologia de Canny [33]. A figura 2.6 mostra o fluxograma do algoritmo desenvolvido. Na figura 2.7 mostra-se resultados da aplicação deste algoritmo. Maiores detalhes da estrutura deste algoritmo podem ser encontrados em [36].

2.2.3 Rastreamento de uma Região de Interesse

Basicamente, este tipo de rastreamento consiste em armazenar os valores dos pixels de uma região de interesse contendo o alvo a ser rastreado. Esta área define a região de referência (*Region of Interest - ROI*) e corresponde ao padrão a ser localizado nas imagens subsequêntes da seqüência.

Em [37] é apresentada esta técnica voltada para determinação do fluxo ótico e do campo de movimento para uma seqüência de imagens com objetos em movimento. Esta técnica rastreia o objeto móvel pela a similaridade, em termos de intensidade dos pixels, da região de interesse onde encontra-se o objeto. A similaridade é medida pela minimização ou maximização de funções de custo.

Como parte do trabalho de pesquisa desenvolveu-se um algoritmo de rastreamento [38]

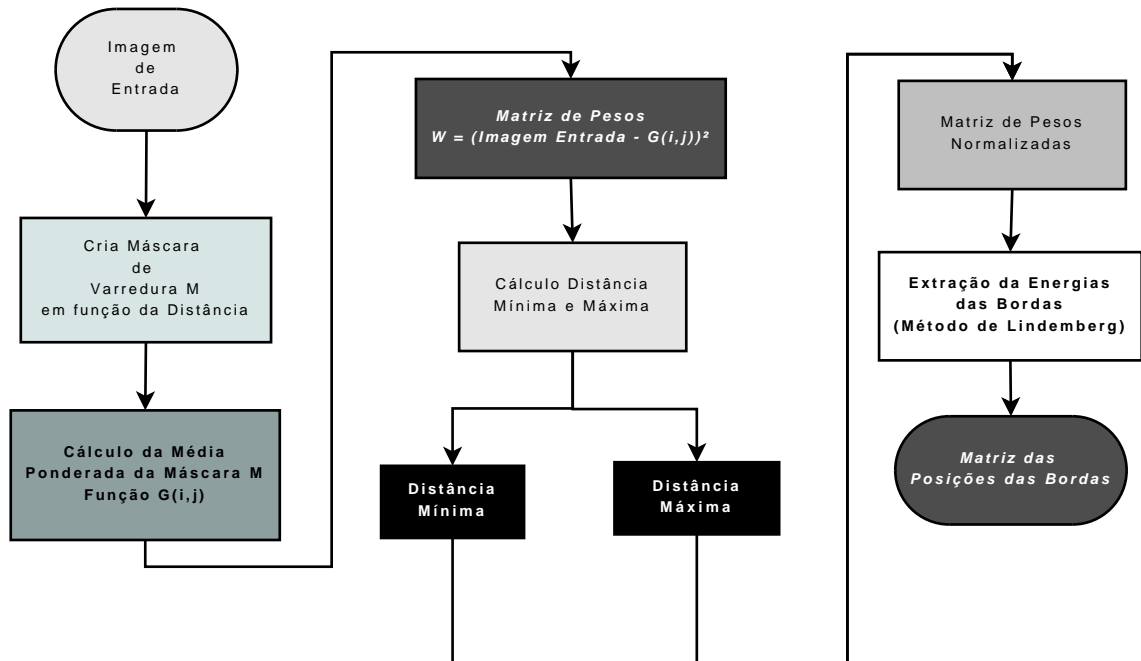


Figura 2.6: Fluxograma do Rastreamento por Bordas utilizando Energia da Imagem.

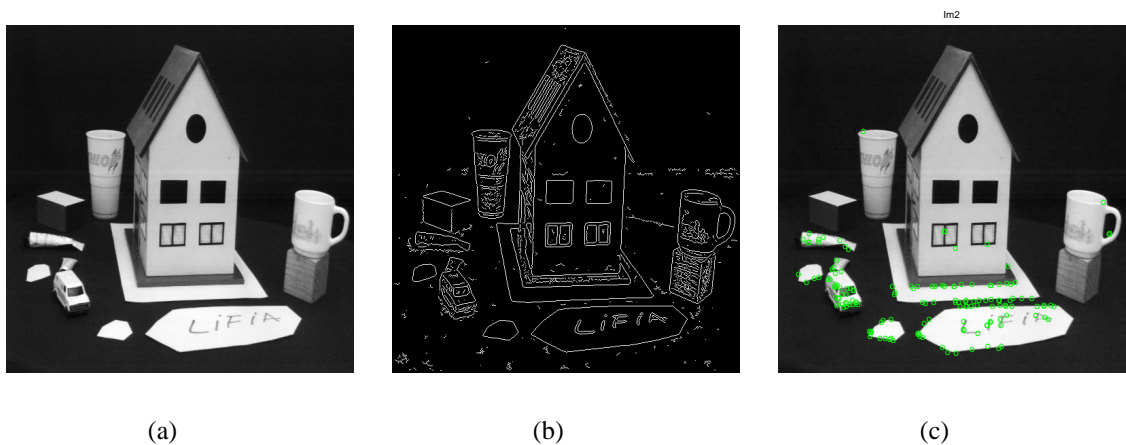


Figura 2.7: Rastreamento por extração de bordas da imagem. Em (a) a imagem original, (b) extração dos limites utilizando a metodologia de Canny e em (c) o resultado do rastreamento das bordas.

baseado na similaridade de regiões, cujo fluxograma é mostrado na figura 2.8. Sua aplicação ao rastreamento de um manipulador robótico é então mostrada na figura 2.9. Entretanto, a maioria dos métodos baseados no rastreamento da *ROI* demanda carga computacional elevada. Em [31] e [39] são apresentados procedimentos para reduzir esta carga.

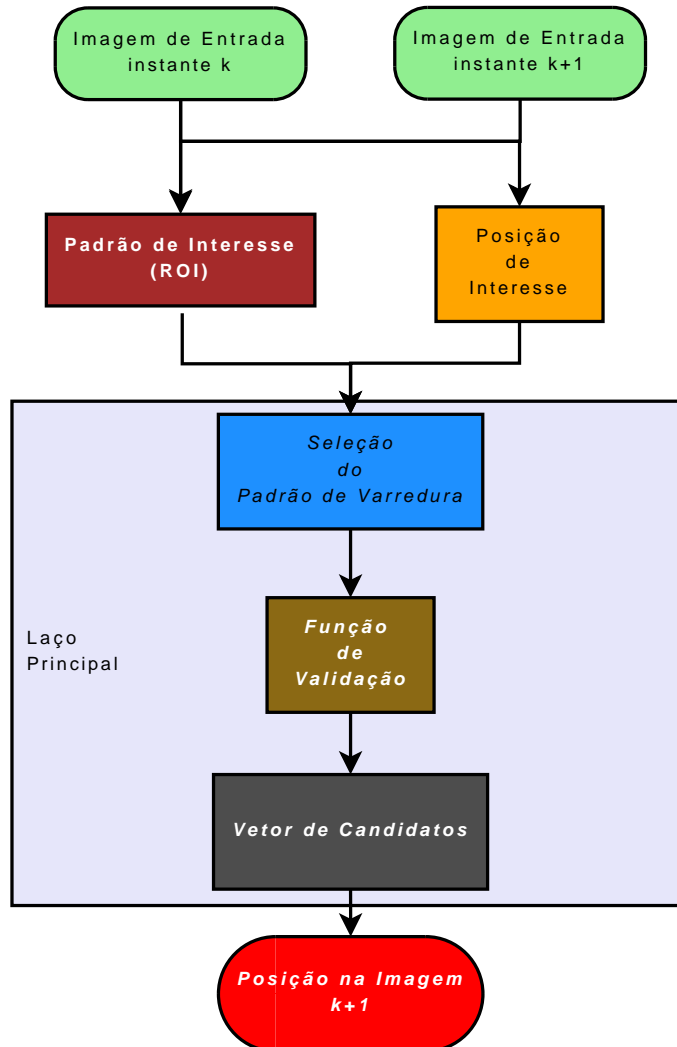


Figura 2.8: Fluxograma do Rastreamento por região de interesse na forma Padrão.

As metodologias que utilizam regiões de interesse são susceptíveis a diversas situações como rotação, mudanças de escala e variações bruscas na iluminação do ambiente. Para casos que envolvam estas características, em [21] propõem-se melhorias baseadas em métodos paramétricos de avaliação do movimento, que minimizam estes efeitos.

Nas seções subseqüentes, os métodos que envolvem rastreamento visual por região de interesse, serão novamente abordados e avaliados com maior detalhamento.

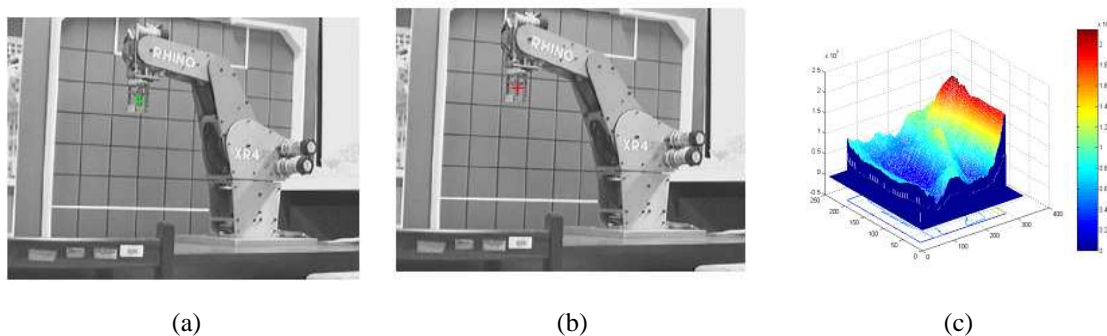


Figura 2.9: Em (a) a imagem no instante k com a região de interesse formada em torno do ponto marcado em verde e em vermelho em (b) a imagem no instante $k + 1$ o pixel correspondente. (c) é a matriz de similaridade da função de custo para a região de interesse em (a).

2.3 MÉTODOS ESTOCÁSTICOS

Os métodos de rastreamento visual utilizando procedimentos estocásticos podem ser agrupados em: *Métodos por Estimação de Estados e Métodos por Associação de Dados*. Vale ressaltar que devido à grande variedade de métodos estocásticos existentes, esta classificação servirá somente como guia para a estruturação das técnicas mais comuns encontradas na literatura.

2.3.1 Rastreamento por Estimação de Estados

Diversos sistemas de rastreamento envolvem a estimação de estado do alvo de interesse. Este processo pode ser dividido em duas partes:

- *Predição de Estado* - obtenção do estado previsto a partir do estado anterior estimado para o quadro atual;
- *Atualização das Medições* - incorpora uma nova medição para o estado estimado para obter a nova estimativa.

A figura 2.10, ilustra a metodologia por estimação de estados. A escolha do modelo de alvo a ser rastreado, representa o fator principal para esta modalidade de rastreamento. Este processo de escolha impõe maior complexidade na presença de alvos que realizam mudanças em sua trajetória [40]. Dentre as metodologias que realizam rastreamento por estimação de estados pode-se destacar duas: Aplicação do *Filtro de Kalman* e o *Filtro de Interação por Múltiplos Modelos - IMM*.

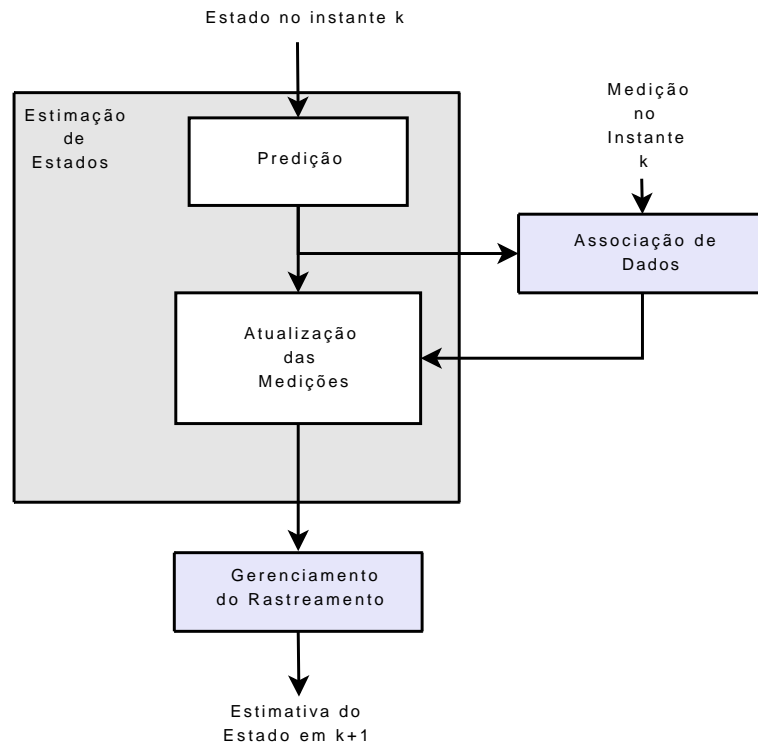


Figura 2.10: Diagrama da Metodologia de rastreamento por estimativa de estados.

2.3.1.1 Filtro de Kalman

O estimador de estado mais utilizado é o *Filtro de Kalman* [10]. O Filtro de Kalman possui desempenho considerado ótimo com a prerrogativa da utilização do erro quadrático médio. Maiores detalhes sobre a formulação, implementações e avaliações com respeito a aplicação do Filtro de Kalman para rastreamento visual, bem como suas formas básicas e transformações, serão abordados no capítulo 4.

2.3.1.2 Filtro de Interação por Múltiplos Modelos

Em diversas situações de rastreamento, os alvos rastreados ocasionalmente realizam manobras de mudança de trajetória. Ao avaliar o procedimento com a incorporação do filtro de Kalman, assume-se um simples modelo para o movimento, dessa forma depara-se com dificuldades impostas por tais mudanças. Um estimador importante para estas situações é o *Filtro de Interação por Múltiplos Modelos - IMM* [41]. O filtro IMM utiliza diversos modelos para o movimento e mudanças probabilísticas dos modelos associados ao alvo. Durante cada período de amostragem (entenda-se como o intervalo entre dois quadros consecutivos), assume-se que o movimento do alvo é descrito por um desses modelos.

O processo de construção do filtro IMM utiliza múltiplos filtros paralelos, em que cada um dos filtros corresponde a um dos modelos assumidos. Devido às diferenças entre os modelos, ocorre uma troca de informações entre os filtros. Durante cada período de amos-

tragem, todos os filtros IMM estão em operação. O estado estimado geral do processo de rastreamento é a combinação das estimativas advindas de cada filtro individualmente. A figura 2.11 apresenta a estrutura da metodologia do filtro IMM[41].

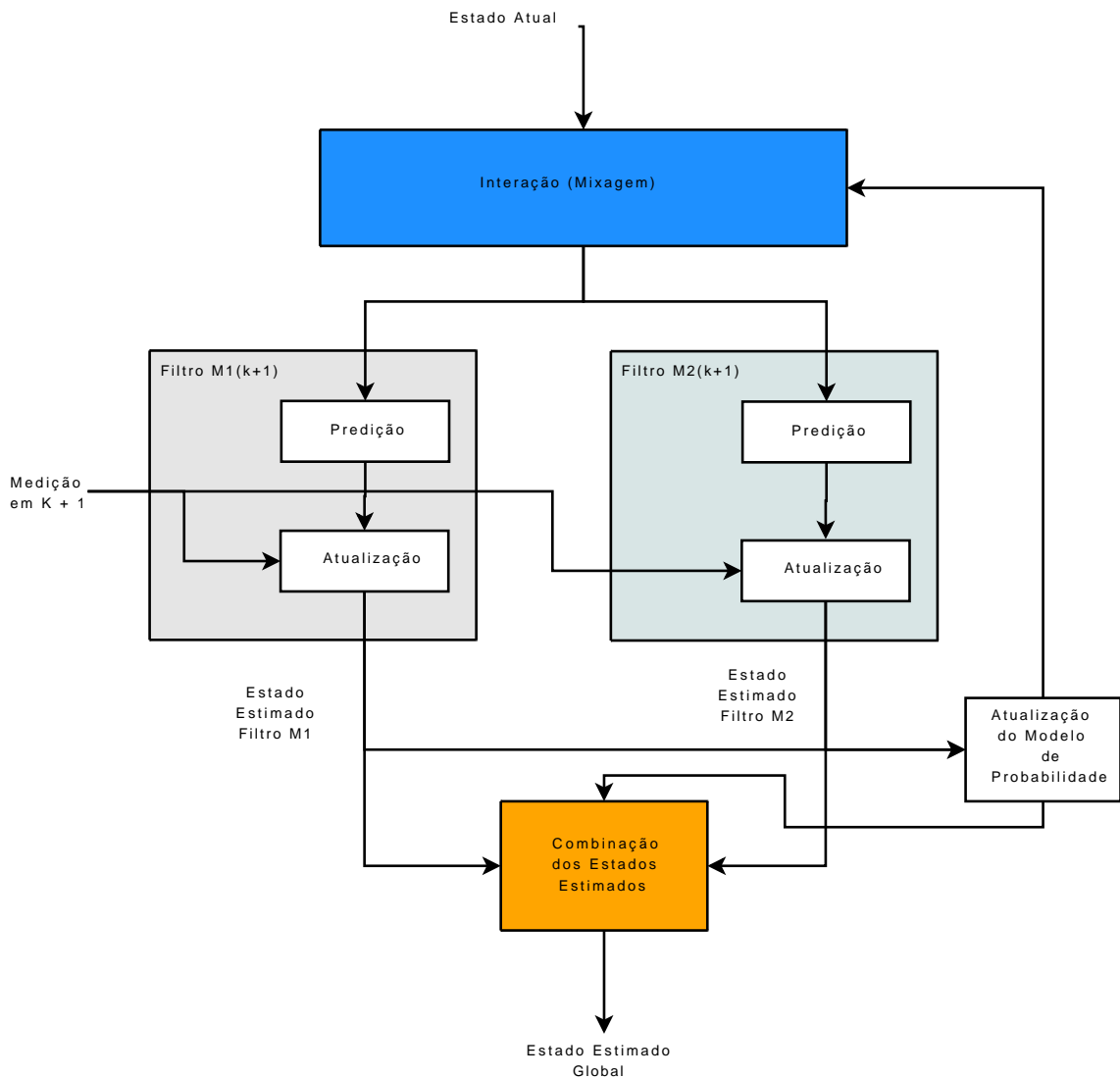


Figura 2.11: Fluxograma da Metodologia de rastreamento por filtro IMM utilizando dois modelos para estimação dos estados.

2.3.2 Rastreamento por Associação de Dados

Diversos algoritmos de rastreamento por associação de dados são apresentados em [42]. Estes algoritmos podem ser definidos como: *Procedimento por Bloqueamento*, *Filtro de Vizinhança-próxima*, *Filtro de Associação Probabilística Conjunta de Dados* e *Filtro de Rastreamento por Múltiplas Hipóteses*.

A associação de dados é definida como o estabelecimento do processo de medição, para qualquer valor utilizado no estimador de estado. Para cada instante de amostragem, assume-

se a existência de conjuntos de rastreadores. Contudo, um conjunto de medições sempre são recebidos. Estas medidas são utilizadas para atualização ou na inicialização de novos rastreadores.

2.3.2.1 Procedimento por Bloqueamento - *Gating Procedure*

O procedimento por bloqueamento [43] é utilizado para a seleção das medições que são candidatas para a associação do rastreamento considerado. Este procedimento permite acelerar o processo de associação de dados pela limitação da região de procura, para as medições provenientes do alvo de interesse no espaço de medição.

Este procedimento também é definido como uma *Hiperelipsóide M-dimensional* centrada sobre a medida predita. O bloqueio estimado para cada rastreador no quadro $k + 1$ é representado por uma elipse centrada na posição prevista e nas medições. Qualquer medida que pertença ao bloqueio é chamada de *medição válida*. Para realizar uma aproximação mais eficiente, requer a utilização do algoritmo de *Jonker-Volgenant* [44], que possui como objetivo uma solução global ótima.

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

De acordo com [29], o filtro de Vizinhança-próxima(NN) é uma variação do método por bloqueamento 2.3.2.1, em que em cada rastreamento o filtro inicia com o método de *Gating*. Então, realiza-se a escolha pela estimativa da medição próxima para medição predita, de acordo com o quadrado do processo de inovação normalizado. Se está medição for válida para mais de um rastreamento, o estado do alvo é atualizado com a medição selecionada.

2.3.2.3 Filtro de Associação Probabilística Conjunta de Dados

No caso de sistemas de rastreamento com alvos próximos, existe um crescimento potencial das associações entre o rastreamento existente e as medições [42]. Sob estas circunstâncias foi proposto o filtro de Associação Probabilística Conjunta de Dados (*Joint Probabilistic Data Association Filter - JPDA*) para resolver esta taxa de falta de correlação existente entre as medições e o rastreamento. De acordo com a abordagem apresentada em [20], a estrutura principal do filtro JPDA é semelhante à do Filtro de Kalman que será abordada com maiores detalhes no Capítulo 4.

A principal diferença encontrada com relação aos filtros JPDA e os anteriormente apresentados, encontra-se na utilização de modelos paramétricos e não-paramétricos [42][45], na obtenção da estimativa de eventos realizáveis, computando as estimativas das probabilidades associadas a cada medição e associação de estados.

2.3.2.4 Filtro de Rastreamento por Múltiplas Hipóteses

O filtro de Rastreamento por Múltiplas Hipóteses (i. e. *Multiple Hypothesis Tracking - MHT*) é uma forma de aproximação para cada tempo de amostragem com a finalidade de realizar a separação das medições nos rastreamentos e alarmes falsos (entenda-se como estimativas aberrantes). Em razão desta abordagem, esta decisão é tomada imediatamente mantendo o recebimento das medições, gerando hipóteses e decisões a sobre esta situação [19]. Conseqüentemente, o número de hipóteses cresce exponencialmente de acordo com o maior número de medições.

De acordo com [19], são gerados inicialmente um número de candidatos que serão rastreados, bem como a realização do processo de recebimento das medições. As hipóteses consistem de rastreamentos cujos estados são estimados, usualmente pelo filtro de Kalman. Para compor este conjunto de hipóteses geradas, utilizam-se também as medições passadas. Novas hipóteses são geradas pela associação dos conjuntos de hipóteses estimadas e as medições recebidas seguindo a ordem de recebimento destas. As possíveis associações que poderão ser realizadas devem obedecer as seguintes condições:

- Se a hipótese é a continuação de um rastreamento já estabelecido;
- Se é um alarme falso e;
- Se é um novo alvo a ser rastreado.

A cada nova hipótese, um rastreamento é associado com a medição mais atual. Pode-se fazer o uso de metodologias como o *Gating Procedure* para reduzir o número crescente de hipóteses, bem como o uso de metodologias de contenção por *Podas (pruning)*, que também realiza o processo de redução do crescimento do números de hipóteses. O método de *Podas* [46] realiza a eliminação de hipóteses levando em consideração as probabilidades associadas a cada hipótese e o limiar máximo permitido de podas.

3 RASTREAMENTO BASEADO NA SIMILARIDADE DE REGIÕES

De acordo com o apresentado na seção 2.1, optou-se inicialmente neste trabalho de pesquisa em realizar o desenvolvimento de algoritmos para rastreamento visual que fossem aplicados em situações diversas. Pôde-se constatar facilmente que os métodos de rastreamento, tanto determinísticos quanto estocásticos, têm suas aplicações são voltadas para situações específicas, tornando-os limitados e de difícil reconfiguração para situações adversas (ex.: rastreamento de objetos que realizam movimentos não-rígidos). Partindo deste pressuposto, surgiu a necessidade do desenvolvimento de algoritmos que sejam adequados para estes tipos de rastreamentos, bem como sejam reconfigurados com maior simplicidade. Desta forma, foi criado o algoritmo *window-matching*, o qual será abordado no decorrer deste capítulo.

Na seção 3.1 descreve-se genericamente o conceito de similaridades de regiões existente na literatura. A seção 3.2 discute os procedimentos de implementação desta metodologia de rastreamento de movimento. Na seção 3.3 apresenta-se um método de rastreamento baseado na similaridade de regiões e que foi desenvolvido neste trabalho. O método proposto é uma versão modificada do método convencional e produziu melhores resultados. A seção 3.4 mostra então o algoritmo desenvolvido que implementa o método proposto. Finalmente a seção 3.5 apresenta os resultados da aplicação deste algoritmo.

3.1 MÉTODOS BASEADOS EM SIMILARIDADE DE REGIÕES

Métodos baseados na medição da similaridade entre regiões em imagens são ferramentas comuns em diversas aplicações como: reconhecimentos de objetos [47], compressão de vídeo [48], correlação entre imagens estéreo [49] e rastreamento por similaridade de regiões (*Window-Matching*) [50]. Muitas aplicações envolvendo similaridade de regiões baseiam-se no emprego de *Funções de Custo* a serem minimizadas ou maximizadas.

O método consiste na análise do padrão do nível de cinza em torno de um ponto de interesse e na busca pelo padrão mais similar na imagem seguinte. Em detalhes, dada uma imagem define-se uma janela $J(x, y)$ em torno de um ponto $p(x, y)$ da região de interesse, consideram-se então janelas similares $J'(x + w, y + h)$ deslocadas horizontal e verticalmente por um número inteiro de pixels que correspondem à largura (w) e altura (h) da janela respectivamente. A similaridade do padrão de cinza entre as janelas é quantificada na forma de uma função de custo. Esta função de custo dependerá das intensidades das duas janelas e é representada como

$$f(J, J'(w, h)). \quad (3.1)$$

Sendo uma medida da dissimilaridade (separação ou distância), a procura da similaridade entre as janelas consistirá então na minimização desta função de custo. No caso da utilização de funções de correlação será a maximização da função.

3.2 ASPECTOS NA IMPLEMENTAÇÃO DOS MÉTODOS DE SIMILARIDADE

As referências [37] e [31] apresentam diversas formas de avaliar as similaridades entre padrões de intensidade de níveis de cinza em imagens sequenciais. Quatro aspectos fundamentais são considerados para a construção dos métodos baseados na similaridade de regiões, são eles: a) *Características do padrão de intensidade de cinza utilizado*; b) *escolha da medida de similaridade (ou função de correlação)*; c) *avaliação da confiabilidade das medidas de similaridade*; e d) *problema da oclusão de objetos na imagem*.

3.2.1 Características do Padrão da Região de Interesse

Nestes métodos de similaridade as informações da rigidez e tipos de movimentos realizados são fatores que determinam fortemente o sucesso ou fracasso da implementação, assim a escolha da região deve ser feita de forma cuidadosa. Caso a região reproduza com alto grau de fidelidade as características atuais da imagem, os resultados obtidos para o rastreamento serão satisfatórios, do contrário não será possível realizar o rastreamento.

Problemas envolvendo aproximações bidimensionais do movimento em imagens, ocorrem quando a região de interesse sofre transformações complexas de forma e iluminação, as quais influenciam diretamente na obtenção da estrutura do movimento (ex: objetos que alteram a profundidade juntamente com uma rotação). Uma das soluções para este problema está na aplicação de métodos que, no decorrer da seqüência de imagens, atualizem a região de interesse a partir da posição anterior da mesma, minimizando os efeitos das mudanças de geometria e iluminação.

Utilizando esta proposta como solução, depara-se com um efeito indesejado conhecido como *Deslocamento do aspecto região de interesse na imagem* (do inglês *Feature Drift*). O efeito de deslocamento da região característica de interesse ocorre quando a nova posição da região de interesse possui um pequeno erro agregado, que se propaga no decorrer da seqüência, interferindo na correta realização do rastreamento. Pode-se minimizar este efeito escolhendo funções para cálculo da medida de similaridade (ou correlação), que sejam adequados com o tipo de seqüência de imagem que será utilizado.

3.2.2 Escolha da Medida de Similaridade

Para realizar o cálculo da medida de similaridade, utilizam-se janelas quadradas de dimensões $N \times N$ para computar o movimento entre uma região centralizada no ponto de interesse $\vec{p}(x, y)$ na imagem I_1 deslocada por valores inteiros i, j (nas direções horizontal e vertical respectivamente), e a imagem I_2 subsequente após realizado um movimento d_x e d_y nas direções horizontal e vertical respectivamente. A seguir algumas funções de custo que podem ser utilizadas para medida da similaridade.

3.2.2.1 Funções de Custo de Similaridade

Em [37] são apresentadas *Funções de Custo* empregadas na correlação entre sinais unidimensionais (ex.: áudio) e bidimensionais (ex.: imagens), elas são mostradas na Tabela 3.1. As funções Soma Absoluta das Diferenças (*Sum of Absolute Differences* - SAD) e Soma do Quadrado das Diferenças (*Sum of Squared Differences* - SSD) são utilizadas em diversas aplicações demandando procedimentos de minimização das funções de custo. Outras aplicações, que utilizam funções de custo, podem ser encontradas nos trabalhos de [51], [52], [53], [54], [55] e [56].

<i>Sigla</i>	<i>Função de Custo</i>
<i>SAD</i>	$\sum_{i,j=-N/2}^{N/2} I_1(x+i, y+j) - I_2(x+i+d_x, y+j+d_y) $
<i>SSD</i>	$\sum_{i,j=-N/2}^{N/2} (I_1(x+i, y+j) - I_2(x+i+d_x, y+j+d_y))^2$
<i>ZSAD</i>	$\sum_{i,j=-N/2}^{N/2} I_1(x+i, y+j) - \bar{i}_1 - I_2(x+i+d_x, y+j+d_y) - \bar{i}_2 $
<i>ZSSD</i>	$\sum_{i,j=-N/2}^{N/2} (I_1(x+i, y+j) - \bar{i}_1 - I_2(x+i+d_x, y+j+d_y) - \bar{i}_2)^2$
<i>LSAD</i>	$\sum_{i,j=-N/2}^{N/2} I_1(x+i, y+j) - \frac{\bar{i}_1}{\bar{i}_2} I_2(x+i+d_x, y+j+d_y) $
<i>LSSD</i>	$\sum_{i,j=-N/2}^{N/2} \left(I_1(x+i, y+j) - \frac{\bar{i}_1}{\bar{i}_2} I_2(x+i+d_x, y+j+d_y) \right)^2$
<i>CC</i>	$\sum_{i,j=-N/2}^{N/2} I_1(x+i, y+j) I_2(x+i+d_x, y+j+d_y)$
<i>NCC</i>	$\sum_{i,j=-N/2}^{N/2} \frac{I_1(x+i, y+j) I_2(x+i+d_x, y+j+d_y)}{\sqrt{\sum_{i,j=-N/2}^{N/2} I_1(x+i, y+j)^2} \sqrt{\sum_{i,j=-N/2}^{N/2} I_2(x+i+d_x, y+j+d_y)^2}}$
<i>ZNCC</i>	$\sum_{i,j=-N/2}^{N/2} \frac{(I_1(x+i, y+j) - \bar{i}_1)(I_2(x+i+d_x, y+j+d_y) - \bar{i}_2)}{\sqrt{\sum_{i,j=-N/2}^{N/2} (I_1(x+i, y+j) - \bar{i}_1)^2} \sqrt{\sum_{i,j=-N/2}^{N/2} (I_2(x+i+d_x, y+j+d_y) - \bar{i}_2)^2}}$

Tabela 3.1: Principais funções de Custo utilizadas na medição da similaridade

Onde:

- N : É a dimensão da região de procura na janela de varredura;
- I_k : Quadro da imagem no instante k ;
- x : Eixo horizontal da imagem;
- y : Eixo vertical da imagem;

- d_x : Deslocamento na direção horizontal;
- d_y : Deslocamento na direção vertical;
- \bar{i}_k : Valor médio dos níveis de cinza da imagem k .

As funções anteriores têm variantes que empregam a eliminação do valor médio (*Zero-Mean SAD - ZSAD* e *Zero-Mean SSD - ZSSD*), bem como a intensidade média local (*Locally SAD - LSAD* e *Locally SSD - LSSD*)¹. As demais funções de custo da tabela 3.1 são aplicadas em outros procedimentos e demandam a maximização (função de Correlação-cruzada (CC), Correlação-cruzada normalizada (NCC) e Correlação-cruzada normalizada com média zero (ZNCC)).

Algumas hipóteses são utilizadas na medida de similaridade. Considera-se que o padrão da intensidade do nível de cinza é aproximadamente constante entre duas imagens sucessivas, principalmente em casos em que não ocorrem variações de perspectiva (escala) ou mudanças de iluminação.

Assume-se também que a textura da região de interesse não possui elevado grau de ambigüidade (ou semelhança) com outras regiões da imagem. Verifica-se que o método oferece melhor desempenho se a textura não é relevante, bem como em casos onde variações de brilho, causadas pelo problema de *aliasing* entre imagens que compõem a seqüência, dificultam o rastreamento por técnicas diferenciais [31].

Um dos problemas dos métodos de similaridade é o esforço computacional necessário para a implementação da técnica, sobretudo para implementações em tempo real [57]. Problemas como o erro de quantização e otimização da técnica serão discutidos mais adiante.

Existem outras variedades de metodologias para realizar o processo de avaliação da similaridade entre as imagens. Entre elas, procedimentos que utilizam: a *Transformada de Rank* [58]; a *Transformada Census* [59], [60]; a *Distância de Hamming* [18]; e a *Distância de Hausdorff* [61] para imagens binárias. Entretanto, o emprego da função de custo, para medida de similaridade em imagens, tem apresentado maior flexibilidade e aplicações mais eficientes. Neste trabalho, optou-se pela *SSD*, que será abordada com maiores detalhes na Seção 3.3, na qual justifica-se a sua utilização para rastreamento visual.

3.2.3 Confiabilidade das Medidas de Similaridade

Na aplicação da função *SSD*, equação 3.2, como medida da similaridade, freqüentemente depara-se com o problema de não ser possível solucionar o problema de correspondência entre regiões com pouca informação de textura. Por esta razão, deve-se utilizar a avaliação

¹Deve-se observar que por serem estruturadas a partir das funções *SAD* e *SSD* originais, estas são empregadas para processos que envolvam a minimização da função de custo

da confiança dos resultados obtidos no cálculo da medida de similaridade, para controlar o grau de influência do erro gerado durante a implementação do método.

$$R(x, y) = \sum_{i, j = -\frac{N}{2}}^{\frac{N}{2}} (I_1(x + i, y + j) - I_2(x + i + d_x, y + j + d_y))^2. \quad (3.2)$$

Em [28], é apresentada uma metodologia de avaliação da confiança dos resultados obtidos para medidas de similaridades que utilizam a função *SSD*. Neste caso, utilizam-se regiões com dimensões 5×5 pixels em comparação com os pontos candidatos a satisfazerem as condições de minimização da função *SSD*. A validação do método parte do princípio que ao realizar a correspondência entre regiões em uma determinada direção de varredura, onde haja uma diferença mínima da distância de similaridade entre os pontos candidatos, não será possível determinar a região correspondente. Em caso contrário, ou seja, existirem variações razoáveis da medida de similaridade em torno daquela direção de varredura, poder-se-á concluir que a região correspondente é aquela que melhor atender aos resultados da função de similaridade.

Neste trabalho determinou-se um mapa de confiabilidade da medida de similaridade. Determinou-se então o mapa de confiabilidade no rastreamento do movimento da ferramenta de um robô manipulador, o resultado é mostrado na figura 2.9 da seção 2.2.3. O mapa de confiabilidade expressa indiretamente no domínio da espacial da imagem vales e picos. A posição destes vales e picos indicam os valores em que a função de custo tem os seus máximos e mínimos locais. Pela localização destes picos e vales pode-se determinar os possíveis pontos candidatos de maior similaridade com relação a região de interesse.

3.2.4 Oclusão de Objetos

Quando simultaneamente rastreia-se vários objetos, com movimentos independentes, aumenta a possibilidade da região de interesse sofrer oclusões. Conseqüentemente, alguns objetos podem desaparecer parcial ou completamente em algumas imagens. Este fato resultará em erros na descrição da trajetória.

Para solucionar o problema da oclusão, pode-se considerar a adição de restrições locais para a trajetória, e/ou trajetórias incompletas podem ser permitidas, desde que esta última abordagem seja devidamente tratada. Uma trajetória incompleta pode ser referida como uma oclusão, aparecimento ou desaparecimento de um objeto, ou também, referida como a perda do ponto de referência do objeto devido a mudança de aspecto, resultante de um movimento ou simplesmente devido a falhas no algoritmo de rastreamento, enfatizando os argumentos da seção 3.2.1.

3.3 UM MÉTODO DE RASTREAMENTO BASEADO EM SIMILARIDADES DE REGIÕES

Nesta seção apresenta-se um método desenvolvido neste trabalho para o rastreamento baseado na similaridade de regiões. As referências [13] e [31] foram as principais fontes inspiradoras para o método. Para medida da correspondência entre janelas utiliza-se a função de custo SSD, somatória das diferenças quadráticas. O rastreamento consiste na minimização desta função entre imagens subsequentes.

3.3.1 A função Somatória das Diferenças Quadráticas (*Sum of Square Differences - SSD*)

A função de custo SSD foi empregada (Eq. 3.2), como medida de similaridade entre regiões de imagens, devido à sua adequação às metodologias de rastreamento visual. Uma característica diz respeito à sua integração à estrutura de movimento (seguindo as hipóteses apresentadas na seção 3.2) voltadas à *estabilidade e estrutura computacional incremental*. Com respeito à estabilidade, a função SSD possibilita, devido ao termo quadrática, que pequenos valores de similaridade entre regiões passem a ter menor influência no processo de correspondência, diminuindo a quantidade de regiões ambíguas entre as regiões da imagem. Esta característica não é encontrada nas funções com estruturas a partir das SADs. Com relação à idéia de estrutura computacional incremental, pode-se notar que ao se adquirir novas informações com respeito a área de aplicação a função SSD, estas são atualizadas de forma a manter as reais características do movimento, permitindo assim processos que abrangem todo o campo visível da imagem.

A equação 3.2 constituirá neste trabalho a função custo de similaridade, e deverá ser minimizada, logo o procedimento consiste em encontrar na imagem subsequente uma região que seja semelhante àquela da imagem anterior. Aplicando essa metodologia, pode-se alcançar a solução para o rastreamento visual. Entretanto alguns detalhes devem ser considerados para realização do rastreamento visual, seguindo as características apresentadas na seção 3.2. Maiores detalhes da utilização da função SSD para o rastreamento serão apresentados na seção 3.4.

3.3.2 Variações da função SSD - Domínio da Frequência

Algoritmos de rastreamento que empregam a similaridade de regiões (principalmente o algoritmo *Window-matching*), utilizam funções de custo - neste caso o interesse é na função SSD, podendo facilmente ser estendido às demais funções de custo apresentadas na tabela 3.1 - para correlacionar possíveis regiões que se assemelham à determinada região da imagem anterior (padrão da região de interesse), pode-se realizar uma abordagem definindo, independente da função de custo, que todo o processo nada mais é que um processo de

correlação bidimensional.

Algoritmos de rastreamento baseados na similaridade de regiões utilizam funções de custo, como aquelas da Tabela 3.1, onde a formulação da função de custo se dá no domínio espacial da imagem. Independentemente da função de custo utilizada, a procura pela similaridade de regiões entre imagens sequenciais, consiste na solução de um problema de correlação bidimensional.

O problema da correlação bidimensional pode ser resolvido também no domínio da frequência transformando a função de custo pela aplicação da Transformada de Fourier Bidimensional [18]. Como um dos objetivos deste trabalho é o desenvolvimento de um algoritmo de rastreamento, decidiu-se investigar a alternativa de trabalhar no domínio da frequência. Para que seja possível o desenvolvimento da estrutura no domínio da frequência, foi empregada abordagem apresentada em [62].

3.3.2.1 Definição do Problema

Partindo do princípio que algoritmos de rastreamento que empregam a similaridade de regiões (baseado na metodologia *Window-matching*), na verdade utilizam uma função de custo - neste caso o interesse é na função SSD, podendo facilmente ser estendido às demais funções de custo apresentadas na tabela 3.1 - para correlacionar possíveis regiões que se assemelham à determinada região da imagem anterior (padrão da região de interesse), pode-se realizar uma abordagem definindo, independente da função de custo, que todo o processo nada mais é que um processo de correlação bidimensional. Partindo desta abordagem, pode-se reescrever a equação 3.2 como:

$$R(x, y) = \sum_{i, j = -\frac{N}{2}}^{\frac{N}{2}} \rho(x, y) \quad (3.3)$$

Em que ρ é uma função bidimensional que representa a estrutura da função de custo do SSD, têm-se:

$$R(x, y) = \sum_{i, j = -\frac{N}{2}}^{\frac{N}{2}} \rho(d_{x,y}(i, j)) \quad (3.4)$$

onde $d_{x,y}(i, j) = (I_1(x + i, y + j) - I_2(x + i + d_x, y + j + d_y))^2$ é uma função que utiliza valores reais dos pixels da imagem. Sabe-se que imagens digitais, independente do espaço de cores em que se trabalhe, faz por convenção o uso de pixels com valores no intervalo de $[0, 255]$, representando na forma digital valores quantizados de 8(oito) *bits*. Desta forma, com a finalidade de simplificar problemas relativos ao condicionamento numérico, os valores para a função da equação 3.4 são normalizados no intervalos entre $[0, 1]$.

De acordo com [63] a correlação bidimensional pode ser expressa pela equação 3.5.

$$R(x, y) = \sum_{i, j = -\frac{N}{2}}^{\frac{N}{2}} (f(x, y) - g(x + d_x, y + d_y))^2 \quad (3.5)$$

Observa-se, que a equação 3.5 representa uma convolução com estrutura quadrática. Passando para a notação de convolução, têm-se:

$$(f * g)(i, j) = \sum_{i, j = -\frac{N}{2}}^{\frac{N}{2}} f(x, y) g^*(x + d_x, y + d_y) \quad (3.6)$$

onde * representa a função complexa conjugada. Aplicando na equação 3.6 a *Transformada Discreta de Fourier - DFT* - bidimensional, pode-se obter a relação entre a estrutura da função SSD com a correlação bidimensional no domínio da frequência (Eq. 3.7).

$$R(x, y) = F(x, y)G^*(x, y) \quad (3.7)$$

em que $F(x, y)$ representa a região de interesse da imagem no instante k aplicado a transformada de Fourier Discreta bidimensional e $G^*(x, y)$ o complexo conjugado da região de interesse da imagem em $k + 1$, também no domínio discreto da transformada de Fourier bidimensional.

Neste trabalho implementou-se a procura pela similaridade de regiões no domínio da frequência e observou-se o seguinte:

- *Elevado custo computacional* - antes da procura pela similaridade há necessidade de aplicar a Transformada Discreta de Fourier, isto representa uma carga computacional adicional. Se comparado com a função de custo no domínio espacial, o número de operações em ponto flutuante, para uma imagem de dimensões $m \times n$ pixels, passa para $2 \times m \times n$.
- *Rotação da Imagem* - Uma propriedade importante da transformada discreta de Fourier, quando aplicada às imagens, é a imunidade à rotação da imagem [18]. Esta característica implica no aumento de regiões na imagem, que no domínio da frequência, implicam no aumento do grau de ambigüidade entre regiões da imagem, dificultando o processo de minimização da função de custo;

Em vista dessas observações, optou-se pela utilização da função de custo SSD no domínio espacial.

3.4 DESENVOLVIMENTO DE UM ALGORITMO DE RASTREAMENTO BASE-ADO NA CORRESPONDÊNCIA ENTRE JANELAS

Um algoritmo seguindo o método da seção 3.1 foi então desenvolvido, ele implementa o rastreamento de objetos em uma sequência de imagens pela procura da correspondência entre janelas. Para medir a correspondência entre janelas de imagens subsequentes utiliza-se a função SSD, embora um usuário pode escolher uma outra função de custo. O algoritmo foi chamado de rastreamento por correspondência de janelas (em inglês o chamaremos de *window-matching based tracking*, *WM*). Uma versão preliminar deste algoritmo foi apresentada em [14], uma versão mais aprimorada foi posteriormente apresentada em [38]. Uma contribuição importante foi a inserção no algoritmo de uma estagio de avaliação da confiabilidade dos resultados. A Figura 3.1 mostra o fluxograma do algoritmo desenvolvido.

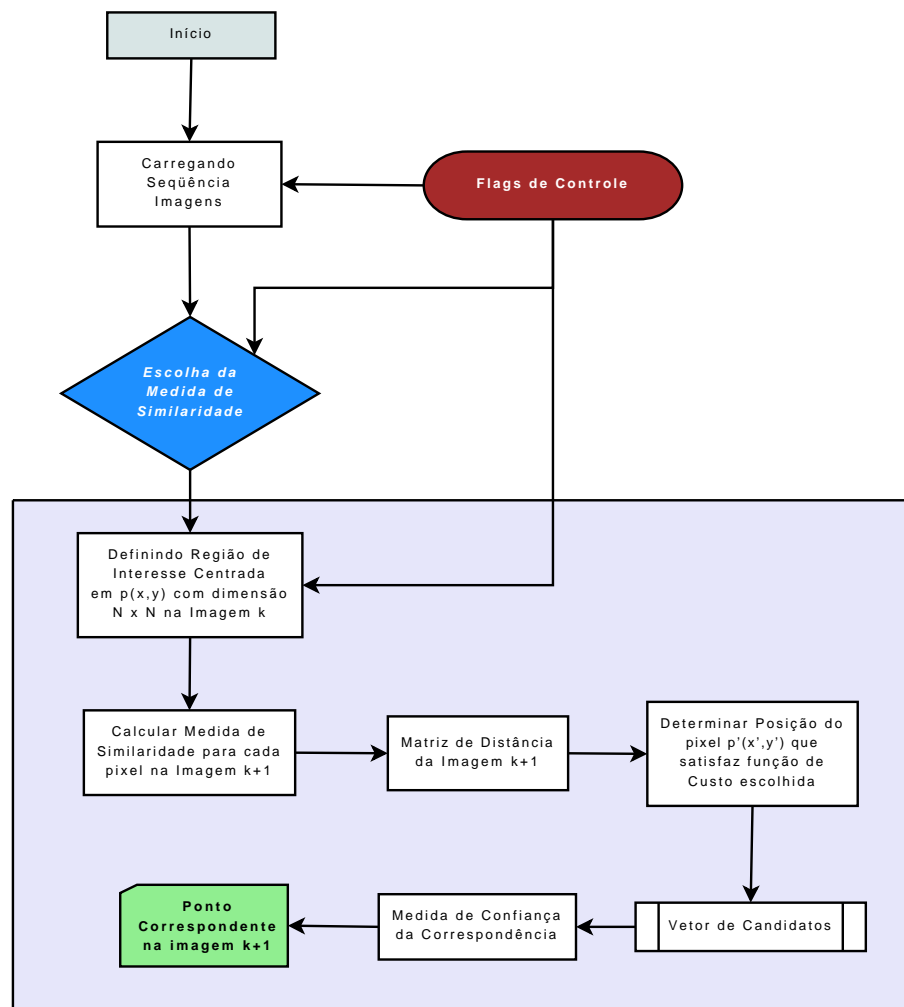


Figura 3.1: Fluxograma do Algoritmo Window-matching

3.4.1 Descrição do Algoritmo WM

As etapas do algoritmo desenvolvido são mostradas no fluxograma da Figura 3.1, são descritas a seguir:

- *Carregamento da Seqüência de Imagens* - O algoritmo deverá ser inicializado com pelo menos duas imagens, seguindo a ordem da seqüência. Na imagem em t_0 é definida a região de interesse, que servirá de padrão para a busca na segunda imagem carregada em t_1 .
- *Escolha da Medida de Similaridade* - O algoritmo permite a escolha da função de custo que será utilizada no processo entre as descritas na tabela 3.1. De acordo com a escolha, o algoritmo opta por um processo de minimização ou maximização.
- *Definição da Região de Interesse (ROI)* - A região é definida em torno de um ponto na imagem do objeto a ser rastreado. Assumindo a hipótese de baixa ambigüidade com outras áreas da imagem, gera-se a máscara do padrão de cinza da ROI que será a referência para a procura da região similar(em termos do padrão de cinza). Deve-se observar também, que este padrão gerado é dinâmico, sendo atualizado a cada nova imagem componente da seqüência. Esta peculiaridade permite melhorias na robustez do rastreamento para situações que envolvam variações bruscas de luminosidade, mudanças de escala (largura, altura e profundidade) das imagens e rotações variadas.
- *Cálculo da Similaridade da imagem k para cada a imagem $k + 1$* - Estabelecido o padrão de referência da ROI na imagem k , o cálculo da similaridade é realizado para cada pixel da imagem $k + 1$. A similaridade resulta do cálculo da função de custo e seu cálculo por pixel demanda evidentemente uma elevada carga computacional. Uma forma de reduzir esta carga é a utilização de sub-regiões de busca com dimensões $r \times s$, sendo $r \leq m$ e $s \leq n$ (onde m e n são respectivamente as dimensões da imagem). As dimensões da sub-região (r e s) são escolhidas de acordo com o tipo e objeto de rastreamento, com isso obtém-se uma redução da área de varredura na qual deve-se calcular a função de custo. Estas sub-regiões de busca podem ser definidas pelo usuário ou determinadas pela inserção, como estágio adicional, de um algoritmo que foi desenvolvido anteriormente nesta pesquisa [27]. Esta última alternativa acrescentaria carga computacional, mas ela pode ser reduzida introduzindo recursos adicionais, tais como, as máscaras de subamostragem, valores pré-calculados da similaridade (*look-up tables*), média ponderada de deslocamento, etc [37].
- *Geração Matriz de Distâncias* - Esta matriz é uma medida da similaridade, quantificada pela função de custo, entre os pixels das imagens k e $k + 1$ das sub-regiões de busca. A matriz de distâncias (de similaridades) tem portanto dimensões $r \times s$. Esta é uma diferença com o método em [37], onde a matriz de distâncias têm dimensões $m \times n$, implicando então em maior carga computacional.

- *Determinação da Posição do Pixel $p'(x, y)$* - Nesta etapa implementa-se um processo de minimização da função de custo(ou maximização, no caso de funções de correlação) para determinação da similaridade entre os pixels. A matriz de distâncias possui vários possíveis pontos de mínimos e máximos locais. De acordo com a hipótese, pela qual assume-se que o movimento entre quadros é limitado, pode-se determinar um mínimo(ou máximo) global. A solução obtida será a posição do pixel $p'(x, y)$, que constituirá a referência para a nova região de interesse.
- *Vetor de Candidatos* - Esta etapa foi inserida para lidar com os casos onde existam regiões com elevado grau de ambigüidade. Também, segundo [64], isto acontece devido aos efeitos de amostragem e quantização na aquisição digital das imagens. Nestes casos a matriz de distâncias pode produzir mais de um ponto de mínimo(ou máximo). O vetor de candidatos está constituído assim de todas estas possíveis soluções. O critério utilizado aqui para selecionar o mínimo(ou máximo) global resulta do cálculo da distância euclidiana [18] para imagens digitais entre o ponto $p(x, y)$ e o ponto $p'(x, y)$ candidato. Independentemente da função de custo escolhida, aquele candidato que tiver menor distância euclidiana será a solução. Nesta decisão aplica-se um método heurístico que segue duas condições: es heurísticas descritas a seguir:
 - i. Verificação se a distância euclidiana entre o ponto atual e o anterior supera a média das distâncias euclidianas calculadas anteriormente; e
 - ii. Caso contrário, atualiza-se a média das distâncias euclidianas entre os pontos atual e anterior.

Estas condições contribuem para diminuir possíveis empates e permitem obter uma única solução, elas foram então implementadas no algoritmo WM, respeitando as hipóteses para o movimento descritas anteriormente.

- *Validação da Correspondência* - Implementou-se no algoritmo WM o procedimento descrito na seção 3.2.3 para avaliar a confiabilidade da correspondência entre os pontos das imagens k e $k + 1$. Esta etapa pode ser ativada pelo *flag* de controle responsável pela compensação do efeito de deslocamento do aspecto região de interesse na imagem, já descrito anteriormente.

Neste algoritmo não é realizada uma etapa de pré-processamento, como as aplicadas em algoritmos anteriores [28] e [31], onde desejava-se realçar algumas características, principalmente das regiões limítrofes de cada objeto na imagem. Testes realizados neste trabalho verificaram que este pré-processamento introduz perturbações desnecessárias na determinação da correspondência entre regiões.

3.5 APLICAÇÕES DO ALGORITMO DE RASTREAMENTO WM

A implementação do algoritmo foi realizada no ambiente computacional científico Matlab, nele utilizou-se o recurso das funções *mexfiles*, com as quais aumenta-se a velocidade de processamento nos cálculos mais críticos em termos de carga computacional. Essas funções demandam a escrita de trechos de programação em linguagem C. O algoritmo WM foi aplicado ao rastreamento de objetos em diversas sequências de imagens. Estas sequências foram cedidas por pesquisadores ou obtidas de universidades. No Anexo B são apresentadas as sequências de imagens utilizadas. Em particular, o algoritmo WM foi aplicado a duas sequências: a) Treinamento de um jogador de tênis de mesa; e b) Tráfego de veículos em um cruzamento de avenidas.

3.5.1 Rastreamento da Bola na Sequência de Jogo de Tênis de Mesa

A sequência (Anexo B.1) mostra a prática de um jogador de tênis de mesa². O problema consiste em rastrear a bola. Nesta sequência a bola realiza movimentos de translação vertical e horizontal, com pouca variação de profundidade e um ambiente uniforme. As dimensões da imagem são de 352×240 pixels no formato *PNG*. A Figura 3.2 mostra alguns quadros do processo de rastreamento da bola realizado pelo algoritmo WM com a função de custo SSD.

Na aplicação do algoritmo a esta sequência utilizou-se uma subregião de busca de 40×40 pixels e uma ROI de 15×15 pixels. O algoritmo fornece então a trajetória da bola. As Figuras 3.3 (a) e (b) mostram a trajetória descrita pela bola nas direções horizontal e vertical respectivamente. Os erros de trajetória na horizontal e na vertical são mostrados nas Figuras 3.4 (a) e (b) respectivamente.

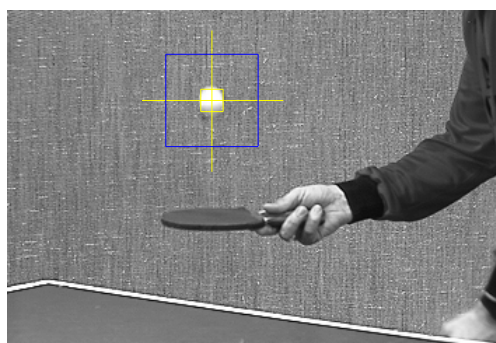
3.5.2 Rastreamento de Veículo em Sequência de Trânsito

A sequência (Anexo B.2) mostra o tráfego de veículos em um cruzamento de avenidas de uma grande cidade³. O problema consiste em rastrear um veículo em particular. Nesta sequência os objetos que compõem a cena realizam movimentos de translação vertical e horizontal, assim como variações de profundidade. O ambiente mostra também outros objetos que poderiam "distrair" o algoritmo. As dimensões da imagem são de 320×240 pixels no formato *JPG*. A Figura 3.5 mostra alguns quadros do processo de rastreamento de um veículo realizado pelo algoritmo WM com a função de custo SSD.

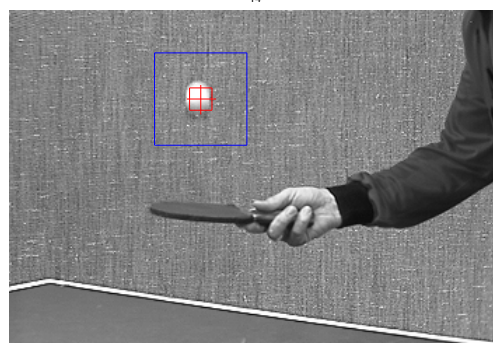
Na aplicação do algoritmo a esta sequência utilizou-se uma subregião de busca de 40×40 pixels e uma ROI de 15×15 pixels. O algoritmo fornece então a trajetória do veículo rastreado. As Figuras 3.6 (a) e (b) mostram a trajetória descrita pelo veículo nas direções

²Sequência obtida em <http://sampl.ece.ohio-state.edu/data/motion/tennis/index.htm>, Ohio State University.

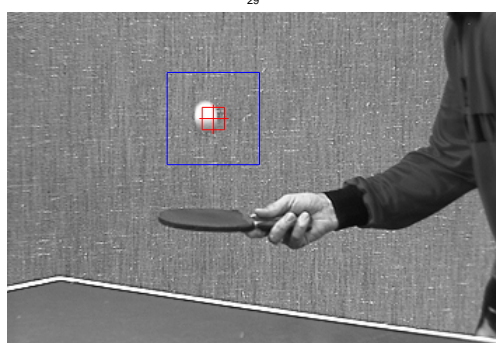
³Sequência gentilmente cedida pelo Prof. H-H. Nagel, Universidade de Karlsruhe, Alemanha.



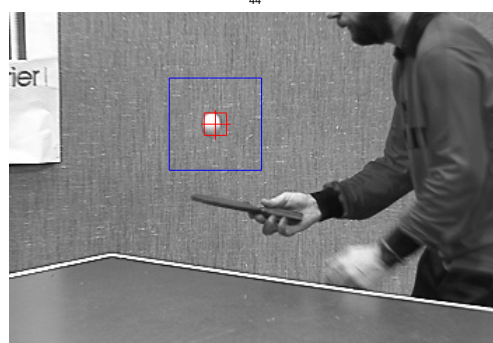
(a)



(b)

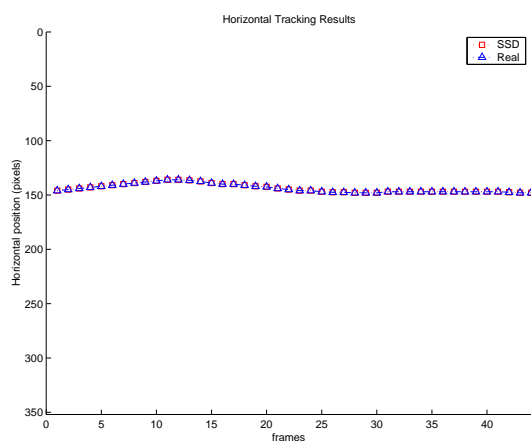


(c)

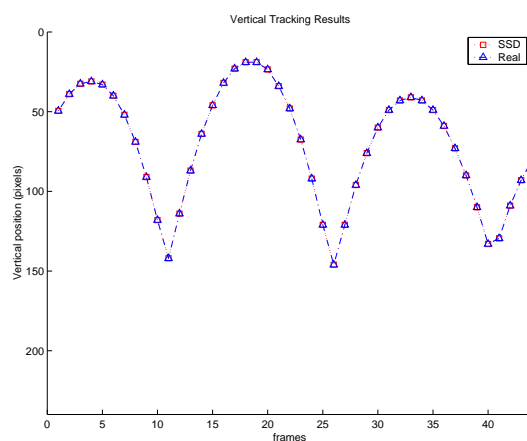


(d)

Figura 3.2: Resultado do rastreamento realizado pelo algoritmo WM utilizando a função de Custo SSD para seqüência do Jogo de Tênis de Mesa.



(a)



(b)

Figura 3.3: Resultado do rastreamento realizado pelo algoritmo WM utilizando a função de Custo SSD para seqüência do jogo de Tênis de Mesa nas direções horizontal (a) e vertical (b). Legenda: \square WM e em \triangle Real

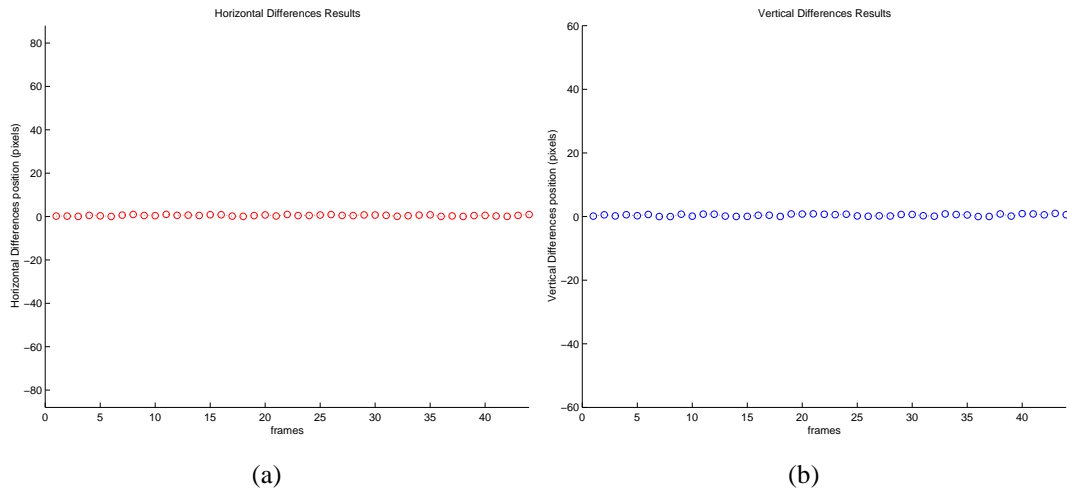


Figura 3.4: Erro de trajetória obtido pelo algoritmo Window-matching em comparação com a trajetória real realizada pelo objeto para a seqüência do jogo de Tênis de Mesa. Em (a) com relação a direção horizontal e em (b) na direção vertical.

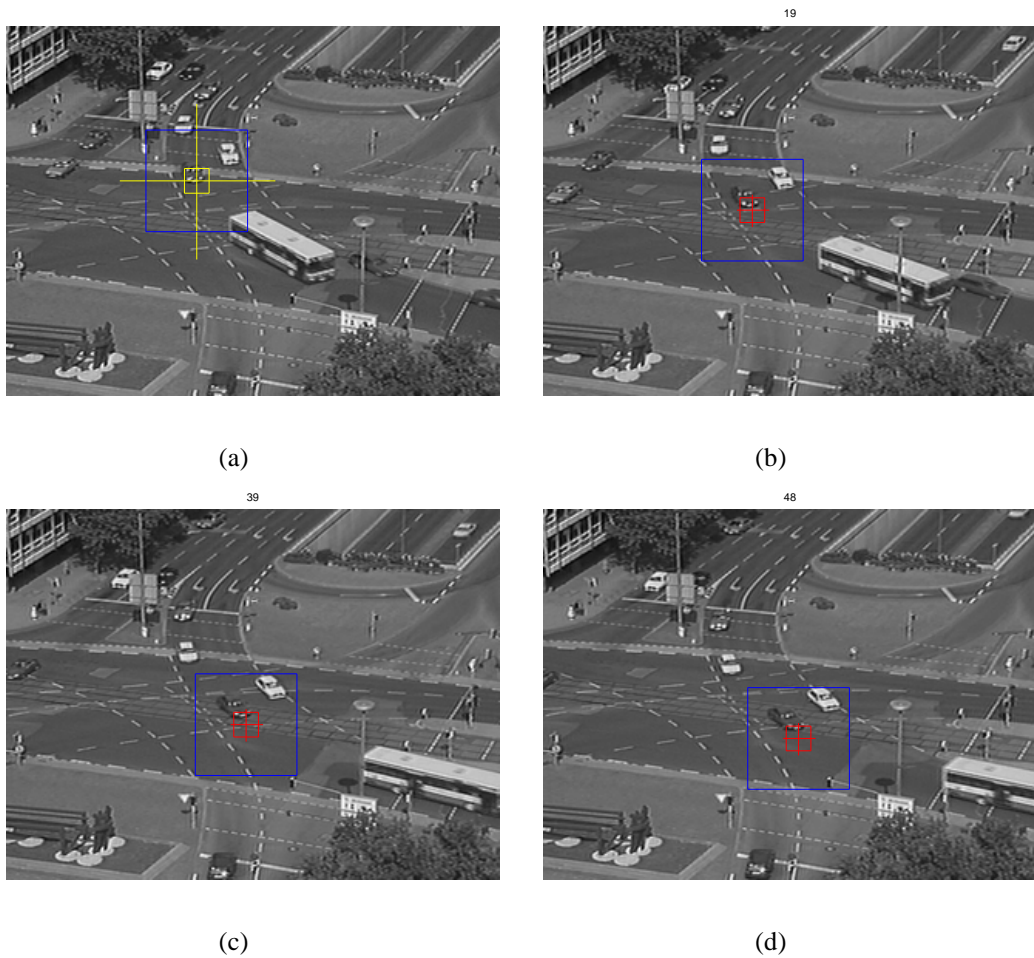


Figura 3.5: Resultado do rastreamento realizado pelo algoritmo WM utilizando a função de Custo SSD para seqüência de Trânsito

horizontal e vertical respectivamente.

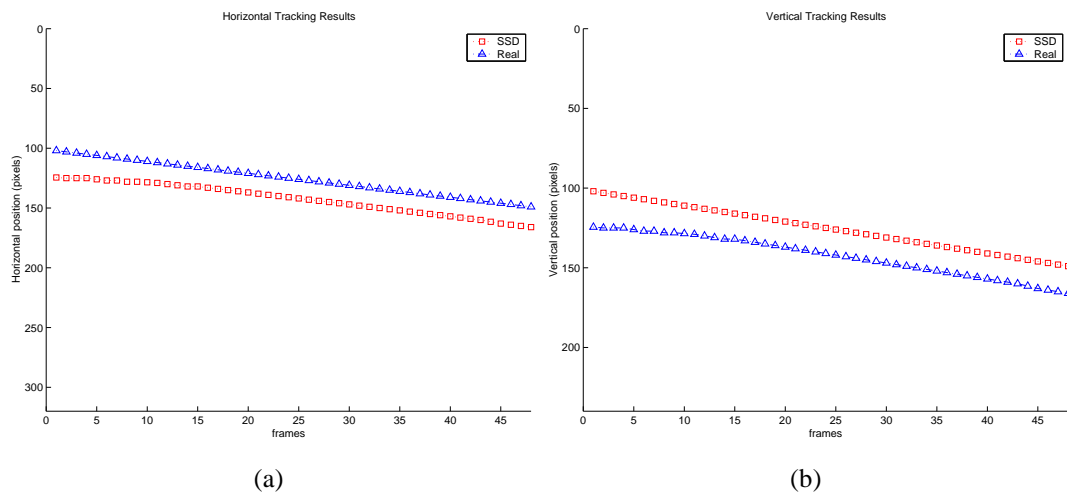


Figura 3.6: Resultado do rastreamento realizado pelo algoritmo WM utilizando a função de Custo SSD para seqüência de Trânsito nas direções horizontal (a) e vertical (b). Legenda: \square WM e em \triangle Real

Os *Erros de Trajetória*, nas direções horizontal e vertical, são apresentados na figura 3.7 (a) e (b). Entende-se como *Erro de Trajetória* a diferença entre as trajetórias reais do objeto e a trajetória apresentada pelo algoritmo de rastreamento em análise.

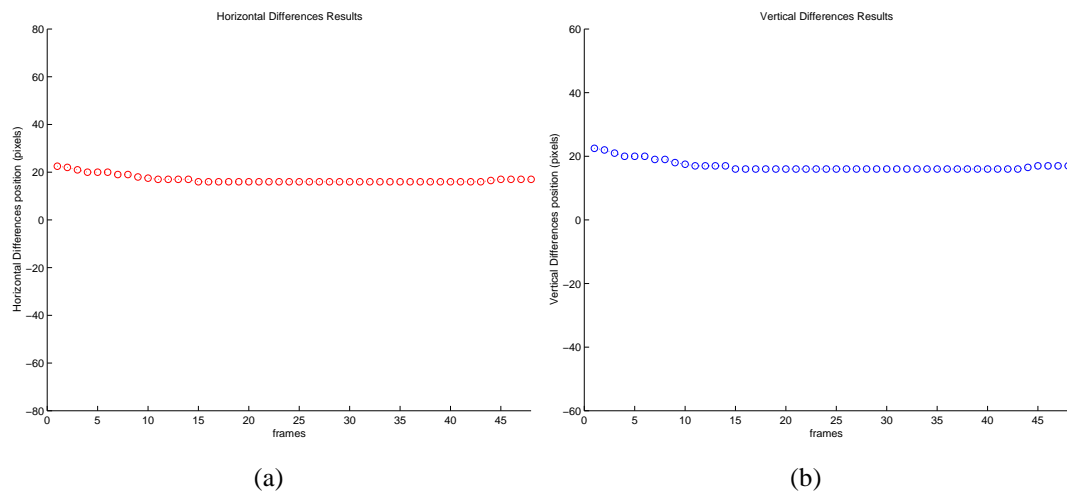


Figura 3.7: Erro de trajetória obtido pelo algoritmo Window-matching em comparação com a trajetória real realizada pelo objeto. Em (a) com relação a direção horizontal e em (b) na direção vertical.

3.6 COMENTÁRIOS SOBRE O ALGORITMO WM

Podem-se tecer agora alguns comentários sobre as características dos métodos de similaridade em geral e como o método de correspondência de janelas, implementado pelo

algoritmo WM e proposto aqui, representa uma contribuição.

Independência da forma do objeto a ser rastreado - Pela utilização de uma janela ROI, o algoritmo WM não sofre influência de variações como mudanças de escala e rotações. Mas o sucesso do algoritmo demanda regiões com baixa ambiguidade.

Demanda ausência de oclusões - Como o rastreamento processa-se em duas dimensões e os movimentos dos objetos acontecem em três dimensões impõe-se uma demanda para o sucesso do algoritmo. A demanda é a ausência de oclusões, ou seja, o objeto rastreado não pode ser oculto por um outro, desse modo a ROI estará sempre visível no plano da imagem.

Dispensa o armazenamento integral da sequência - Diferente dos outros métodos de rastreamento visual, o algoritmo WM dispensa o armazenamento da sequência inteira para processamento. Ele requer unicamente o armazenamento da última imagem. Desse modo, reduz-se tanto o tempo de processamento quanto o espaço de memória necessário.

Busca heurística - A implementação de métodos heurísticos na procura pela similaridade demanda, em geral, aumento da carga computacional. Isso compromete a utilização do algoritmo em aplicações em tempo real. Algumas tentativas de solução serão apresentadas mais adiante.

Efeito de variação de aspectos da imagem (*feature drift*) - Procedimentos para reduzir este efeito, descrito na subseção 3.2.1, será apresentados no Capítulo 4.

O algoritmo de correspondência de janelas (window matching - WM) oferece a vantagem de ter poucos parâmetros a configurar (comparado aos algoritmos apresentados no capítulo 2) e independência da geometria do objeto, do espaço de cores (padrões), do movimento. O algoritmo pode ser utilizado não somente para rastreamento visual, mas para codificação de imagens digitais, segmentação de movimento, compactação de vídeo.

A utilização de subregiões (janela de busca) em torno da região de interesse reduziu o tempo de processamento e portanto a carga computacional. A dimensão da janela de busca é escolhida de acordo com o objeto a ser rastreado e com os outros elementos que compõem a cena. Também interferem na escolha do tamanho da janela de busca outros fatores como: grau de ambiguidade, variações bruscas de iluminação. Para lidar com estes efeitos, que verificou-se podem levar ao insucesso do algoritmo determinístico, aperfeiçou-se o algoritmo acrescentando uma abordagem estocástica. Isto levou à introdução da filtragem de Kalman, como será visto no Capítulo 4.

4 FILTRAGEM ESTOCÁSTICA NO RASTREAMENTO VISUAL

Este capítulo trata da abordagem estocástica, em particular da filtragem de Kalman, ao problema de rastreamento visual. Na seção 4.1 comenta-se o papel do Filtro de Kalman. A seção 4.2 segue com uma breve descrição do Filtro de Kalman Discreto. A aplicação deste filtro ao rastreamento visual é tratada na seção 4.2.2, onde apresenta-se um algoritmo de correspondência de janelas com filtragem de Kalman, desenvolvido neste trabalho. O algoritmo foi chamado de algoritmo WM+K. Finalmente a seção 4.3 relata a aplicação deste algoritmo de rastreamento de objetos em situações onde o algoritmo WM não consegue rastrear o objeto. Isto acontece principalmente quando há um alto grau de ambigüidade na imagem e na presença do efeito de desvio de aspectos da ROI (*feature drift*).

4.1 FILTRO DE KALMAN

Em 1960, R. E Kalman publicou seu famoso artigo, em que apresentou uma solução recursiva para o problema da filtragem linear discreta de dados [10]. Desde então, devido aos grandes avanços na capacidade de processamento de dados digitais, o **Filtro de Kalman** tem sido objeto de extensiva pesquisa e aplicações, particularmente nas seguintes áreas: navegação autônoma ou assistida [65], planejamento de sistemas de transportes [66], detecção de falhas e isolamento de sensores e atuadores em turbinas à gas de aeronaves [67], sistema de controle visual para robôs móveis [68], entre outras dezenas de aplicações que podem ser encontradas na literatura.

O *Filtro de Kalman* oferece uma ferramenta computacional recursiva para realizar a estimação do *estado* de um processo, aplicando como metodologia a minimização quadrática do erro. Através da observação de variáveis denominadas *variáveis de observação*, o filtro permite a estimação das variáveis de estado do processo.

Pode-se afirmar que o filtro é ótimo, em sua forma linear, caso seja possível modelar o ruído de processo e o ruído de medição como ruído Gaussiano Branco ([69] e [70]). O Filtro de Kalman é uma ferramenta poderosa em diversos aspectos: oferece subsídios para estimações do passado, presente, e em alguns casos, permite a estimações de estados futuros e ainda pode realizar estas estimações quando a natureza do sistema modelado não é muito conhecida [70].

4.2 O FILTRO DE KALMAN DISCRETO

A abordagem nesta seção, segue a apresentação e nomenclatura de [71]. Optou-se como referência principal de forma a manter padronizadas as notações. Por se tratar de uma metodologia consagrada, maiores detalhes podem ser obtidos nas seguintes referências: [72], [73] e [74].

4.2.1 Formulação do Problema

O Filtro de Kalman tem como principal objetivo estimar o estado do vetor de estados $\mathbf{x} \in \mathbb{R}^n$ de um processo no tempo discreto, modelado a partir de uma equação linear estocástica a diferenças finitas:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad (4.1)$$

com as medições realizadas por $\mathbf{z} \in \mathbb{R}^m$ dada por

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (4.2)$$

As variáveis aleatórias \mathbf{w}_k e \mathbf{v}_k representam o ruído de processo e o ruído de medição respectivamente, assumidas como independentes (uma da outra), definidas como ruído branco e com distribuição normal

$$\begin{aligned} p(w) &\sim N(0, \mathbf{Q}) \\ p(v) &\sim N(0, \mathbf{R}). \end{aligned} \quad (4.3)$$

Na prática, a matriz de *covariância do ruído de processo* \mathbf{Q} e da *covariância de ruído das medições* \mathbf{R} podem se alterar a cada instante de tempo e/ou medição, entretanto neste caso são consideradas constantes somente para fins de modelagem matemática do processo descrito pela equação 4.1.

A matriz \mathbf{A} (dimensão $n \times n$) na equação 4.1 descreve a transição do estado no instante $k - 1$ para o instante k . A matriz \mathbf{B} de dimensões $n \times l$ relaciona as entradas do processo $\mathbf{u} \in \mathbb{R}^n$ para o estado \mathbf{x} . A matriz \mathbf{H} (dimensão $m \times n$) relaciona o estado às medições obtidas durante a evolução do processo.

Define-se $\hat{\mathbf{x}}_k^- \in \mathbb{R}^n$ como sendo o estado estimado *a priori* no instante de tempo k e $\hat{\mathbf{x}}_k \in \mathbb{R}^n$ como sendo a estimativa *a posteriori* no instante k da medição \mathbf{z}_k . Dessa forma, pode-se definir como erro de estimativa *a priori* e *a posteriori* como

$$\begin{aligned}
e_k^- &\equiv \mathbf{x}_k - \hat{\mathbf{x}}_k^- \\
e & \\
e_k &\equiv \mathbf{x}_k - \hat{\mathbf{x}}_k.
\end{aligned} \tag{4.4}$$

A estimação *a priori* da covariância do erro é

$$P_k^- = E \left[e_k^- e_k^{-T} \right], \tag{4.5}$$

e a estimação *a posteriori* da covariância do erro é

$$P_k = E \left[e_k e_k^T \right], \tag{4.6}$$

Segundo [10] o objetivo é obter uma estimativa *a posteriori* para o estado $\hat{\mathbf{x}}_k^-$ e uma diferença ponderada entre a medição atual \mathbf{z}_k e a medida prevista $\mathbf{H}\hat{\mathbf{x}}_k$ segundo a equação 4.7.

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k) \tag{4.7}$$

A diferença $(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k)$ é chamada de *medida de inovação*. A medida de inovação reflete a discrepância entre a medição predita $\mathbf{H}\hat{\mathbf{x}}_k$ e a medida atual \mathbf{z}_k . Caso o resultado da inovação for zero, significa que a medida estimada e a predita estão em concordância. A matriz \mathbf{K} de dimensões $n \times m$ (Eq. 4.7) é definida como o *ganho* que minimiza o erro em covariância *a posteriori*. A equação 4.7 pode ser obtida manipulando a equação 4.6, obtendo os valores esperados e ainda fazendo o valor de \mathbf{K} igual a zero. Uma forma de obter o resultado para \mathbf{K} que minimiza a equação 4.6, é

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1}. \tag{4.8}$$

onde P_k^- é a matriz de covariância do processo *a priori*. Observando a equação 4.8 pode-se verificar que se a covariância do erro de medição \mathbf{R} aproxima-se de zero, o ganho \mathbf{K} influenciará com maior intensidade a inovação, especificamente:

$$\lim_{R_k \rightarrow 0} \mathbf{K}_k = \mathbf{H}^{-1}. \tag{4.9}$$

Por outro lado, se a estimação *a priori* da covariância do erro \mathbf{P}_k^- tende a zero, o ganho \mathbf{K} influenciará com maior intensidade a inovação, ou seja:

$$\lim_{\mathbf{P}_k^- \rightarrow 0} \mathbf{K}_k = 0. \tag{4.10}$$

4.2.2 O Algoritmo de Filtragem de Kalman

Como visto, a filtragem de Kalman é um procedimento recursivo ótimo para a estimação do estado de um sistema dinâmico, agregando ruídos, de medições e processo, e um modelo estocástico das incertezas do sistema dinâmico. Apresenta-se então a formulação computacional para o algoritmo de filtragem.

O Filtro de Kalman estima o estado de um processo usando realimentação de estados: o filtro estima o estado do processo para um instante de tempo qualquer e obtém por realimentação as medições ruidosas. De uma forma simplificada, pode-se dividir as equações do Filtro de Kalman em duas etapas: **Atualização do Tempo ou Predição** (*Time Update*) e **Atualização das Medições ou Correção** (*Measurement Update*). A etapa de Atualização do Tempo é responsável pela projeção do estado atual e estimativas do erro de covariância para obter a estimativas *a priori* para o próximo instante de tempo. A Atualização das Medições é responsável pelo processo de realimentação, incorporando uma nova medição na estimativa *a priori* para obter uma estimativa *a posteriori* melhorada. A figura 4.1 mostra a correlação entre as etapas de Predição e Correção do Filtro de Kalman.

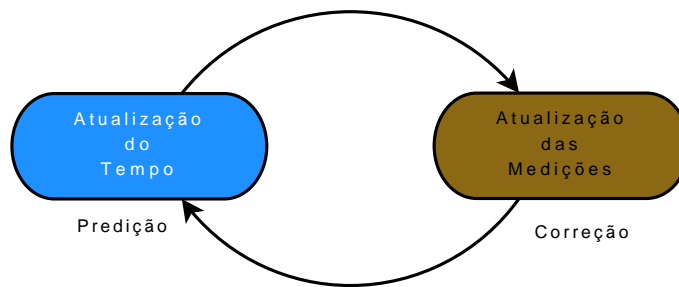


Figura 4.1: Dinâmica da execução do Algoritmo do Filtro de Kalman

As equações que descrevem as etapas de Predição (Eqs.: 4.11 e 4.12) e Correção (Eqs.: 4.13, 4.14 e 4.15) do Filtro de Kalman são:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} \quad (4.11)$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q} \quad (4.12)$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (4.13)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k) \quad (4.14)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H})\mathbf{P}_k^- \quad (4.15)$$

onde \mathbf{I} é a matriz identidade.

A primeira etapa do algoritmo é, durante o processo atualização das medições, obter o ganho \mathbf{K}_k do filtro¹. A etapa seguinte é atualizar a medição do processo para obter a matriz \mathbf{z}_k , e gerar uma estimativa *a posteriori* do estado pela incorporação das medições (Eq. 4.14). A etapa final do algoritmo é obter a covariância *a posteriori* do erro estimada pelo equação 4.15.

Após cada instante de tempo, em conjunto com uma nova atualização da medição, este procedimento se repete com a estimativa *a posteriori* anterior para calcular as novas estimativas *a priori*. Este aspecto recursivo é uma das principais características do Filtro de Kalman, tornando-o na prática com menor complexidade em sua implementação, se comparado com outras modalidades de filtros com aplicações semelhantes (ex.: Filtro de Wiener [75], [18]). No Anexo C são apresentadas os conceitos e as estruturas das principais variações do Filtro de Kalman.

4.3 O FILTRO DE KALMAN APLICADO AO RASTREAMENTO VISUAL

A filtragem de Kalman, pode ser utilizada em aplicações que envolvem rastreamento visual de objetos em imagens seqüenciais para:

1. Estimar a posição de um ponto característico no próximo quadro, procurando regiões de características similares no próximo quadro; e
2. Calcular as incertezas desta estimação, determinando o grau de confiabilidade em encontrar esta característica no próximo quadro na região próxima do ponto previsto.

No rastreamento de objetos quadro a quadro em seqüências de imagens o movimento do ambiente observado é usualmente contínuo, permitindo então realizar a predição (e/ou estimativa) do movimento de pontos da imagem, para qualquer instante, baseado em informações prévias de trajetórias. O rastreamento visual de objetos pode ser então formulado como um problema de estimação de estados de um sistema dinâmico de movimento, definindo como vetor de estados $\mathbf{x} = \begin{bmatrix} x & y & u & v \end{bmatrix}^T$, onde $\mathbf{p} = \begin{bmatrix} x & y \end{bmatrix}^T$ são as posições horizontal e vertical e $\dot{\mathbf{p}} = \begin{bmatrix} u & v \end{bmatrix}^T$ suas velocidades correspondentes. Como um novo quadro da seqüência de imagens é adquirido e processado para cada instante de tempo $t_k = t_0 + k\Delta t$, com $k = 0, 1, 2, \dots$ e Δt é o intervalo de tempo entre os quadros. Dessa forma, para um intervalo de tempo entre quadros pequeno, o vetor velocidade de estados é constante, assim o modelo dinâmico do sistema é descrito por:

¹A equação do ganho do filtro apresentada na Eq. 4.13 é a mesma apresentada na Eq. 4.8.

$$\begin{aligned}
x_k &= x_{k-1} + u_{k-1}\Delta t \\
y_k &= y_{k-1} + v_{k-1}\Delta t \\
u_k &= u_{k-1} \\
v_k &= v_{k-1}
\end{aligned} \tag{4.16}$$

Adotando-se² $\Delta t = 1$, e incluindo nas equações de estado o vetor \mathbf{w}_{k-1} para representar o ruído de processo, a equação de estado será:

$$\mathbf{x}_k = \Phi_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{4.17}$$

Neste trabalho o filtro de Kalman atuará como "supervisor" do algoritmo de correspondência de janelas WM (*Window-matching*). O vetor de posições \mathbf{p} , componente do vetor de estados \mathbf{x}_k será fornecido para o filtro de Kalman pelo algoritmo WM. Acrescentando o ruído de medição de posições \mathbf{v}_k , o vetor de medições \mathbf{z}_k do filtro será então:

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_k \tag{4.18}$$

Conforme já declarado na seção 4.2, os vetores de ruído \mathbf{w}_k e \mathbf{v}_k são assumidos como tendo distribuição Gaussiana com média zero. As matrizes de covariância do processo \mathbf{Q}_{k-1} e de covariância das medições \mathbf{R}_{k-1} são consideradas entradas do Filtro de Kalman no instante de tempo t_{k-1} .

Para inicializar o processo de estimação são utilizados valores elevados para a matriz de covariância do erro \mathbf{P}_0 devido ao filtro levar em consideração o nível de confiança das estimativas baseado nesta matriz. Em muitos casos, estimações indesejadas podem ser obtidas devido a problemas de condicionamento numérico da matriz \mathbf{P}_0 e/ou dificuldades no processo de convergência dos valores da atualização da matriz \mathbf{P}_0 , gerando problemas na obtenção no cálculo do ganho e conseqüentemente polarizando o filtro.

Como no rastreamento em imagens o movimento é modelado por equações lineares, optou-se em utilizar as equações do algoritmo do Filtro de Kalman linear (seção 4.2.2). A estimação ótima é dada pelo vetor de saída $\hat{\mathbf{x}}_k$ no instante de tempo t_k , representando a posição na imagem e sua respectiva velocidade, tendo suas incertezas representadas pelo elementos da diagonal da matriz \mathbf{P}_k .

²Para aplicações em tempo real Δt deve ser ajustado convenientemente.

4.3.1 Rastreamento Visual por Correspondência de Janelas com Filtro de Kalman - Algoritmo WM+K

Para incrementar seu desempenho o algoritmo WM(Figura 3.1) foi então modificada pela inserção de um estágio de filtragem de Kalman, o novo algoritmo foi chamado de algoritmo WM+K. Enquanto o algoritmo WM está em execução, o processo de filtragem de Kalman processa os resultados das medições gerando saídas para indicar as medidas de tolerância. Caso o algoritmo WM retorne valores que não são compatíveis com as estimações oriundas do Filtro de Kalman, esses resultados são descartados e as estimações providas pelo Filtro de Kalman são utilizadas como novas referências no rastreamento para o próximo quadro. Com esta estratégia conseguiu-se obter menores índices de falhas no rastreamento, bem como uma melhoria mínima no efeito de *drift*.

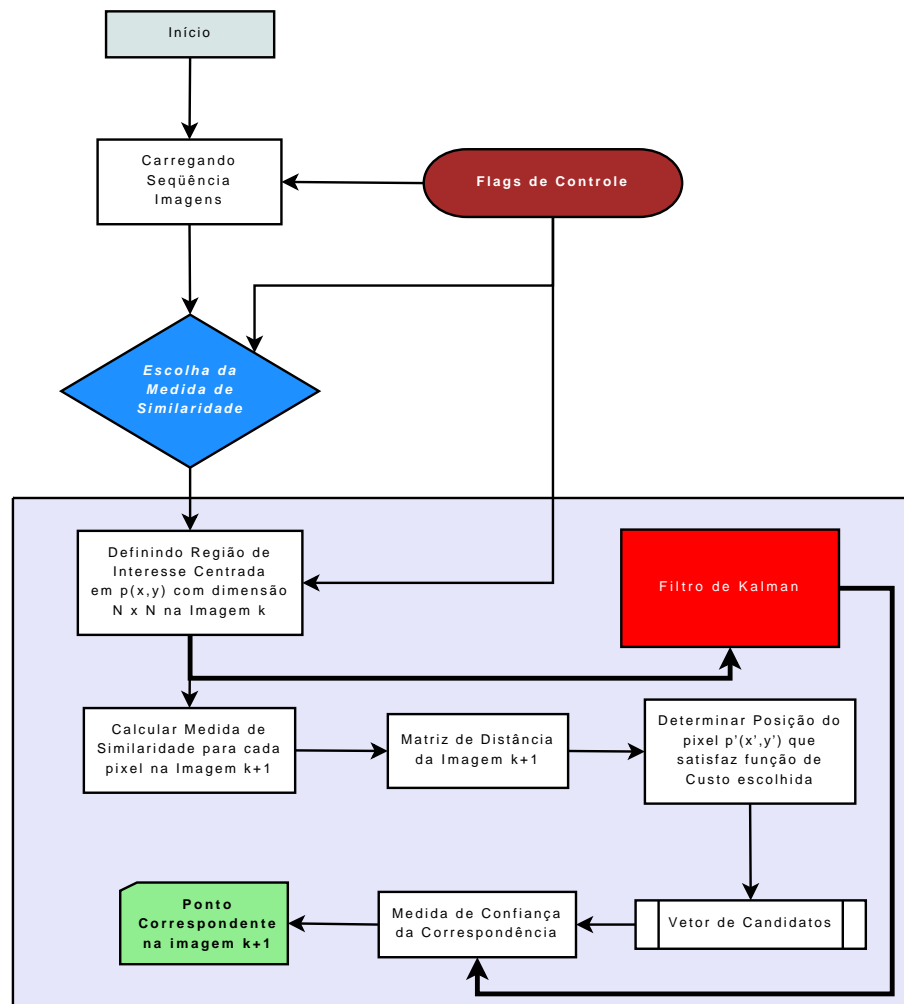


Figura 4.2: Fluxograma do Algoritmo Window-matching com etapa de Filtragem de Kalman (WM+K)

A figura 4.2 mostra a estrutura do algoritmo WM+K, onde observa-se o estágio de filtragem de Kalman. O filtro processa as informações provenientes do algoritmo WM. Após realizada a etapa do algoritmo WM e de filtragem de Kalman, o estágio de Medida da Confi-

ança e Tolerância faz a verificação dos valores retornados por ambos os processos (Window-Matching e Filtro de Kalman), observando de forma simples (média aritmética entre as posições vertical e horizontal), o grau de discrepância entre o valor obtido pelo WM e as estimativas do Filtro de Kalman. Caso seja superior a um valor de tolerância pré-determinado pelo usuário, o sistema avalia com relação à medida anterior proveniente de cada algoritmo. Aquela que possuir maior discrepância será atualizada pela medida do outro. Com este acoplamento e estratégia de decisão, o nível de robustez e eficiência do algoritmo WM+K na realização do rastreamento visual aumentou de forma considerável propiciando a diminuição de rastreamentos errôneos e/ou não localização da região de interesse. O algoritmo WM+K foi então aplicado ao rastreamento de objetos em seqüências nas quais o algoritmo WM apresentou deficiências.

4.4 APLICAÇÕES DO ALGORITMO DE RASTREAMENTO WM+K

A implementação do algoritmo WM no ambiente Matlab foi então modificada para incluir o estágio de filtragem de Kalman. O algoritmo WM+K resultante foi então aplicado ao rastreamento de objetos em duas seqüências: *a) Encontro e caminhada de pessoas em local público*; e *b) Garrafa flutuando no mar*. As seqüências de imagens estão no Anexo B.

4.4.1 Rastreamento de Indivíduos em Seqüências de Circulação de Pessoas em Locais Públicos

Em locais públicos existem geralmente câmeras de vigilância monitorando a circulação de pessoas. Para aplicação do algoritmo de rastreamento WM+K, duas seqüências envolvendo a circulação de pessoas são consideradas.

*Caso 1 - Encontro e caminhada de duas pessoas em prédio público*³

A seqüência (Anexo B.4) mostra duas pessoas se aproximando, cumprimentando-se e caminhando juntas. As cenas apresentam distorções fométricas, ruídos de naturezas diversas (amostragem, quantização e influência da iluminação), regiões com grau apreciável de ambigüidade e mudanças de escala. As dimensões das imagens são de 348×288 pixels no formato *JPG*. Neste caso uma subregião de busca de 40×40 pixels e uma ROI de 10×10 pixels mostraram-se adequadas.

O problema de rastreamento trata de acompanhar a trajetória de um dos indivíduos na seqüência. Para efeito de comparação, aplicou-se primeiro o algoritmo WM de rastreamento por correspondência de janelas sem filtragem estocástica. A figura 4.3 mostra alguns quadros do processo de rastreamento do indivíduo, nos quais a ROI aparece na cor vermelha. Observa-se que o algoritmo não conseguiu acompanhar o indivíduo no seu deslocamento.

³Seqüências de imagens obtidas em <http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/>

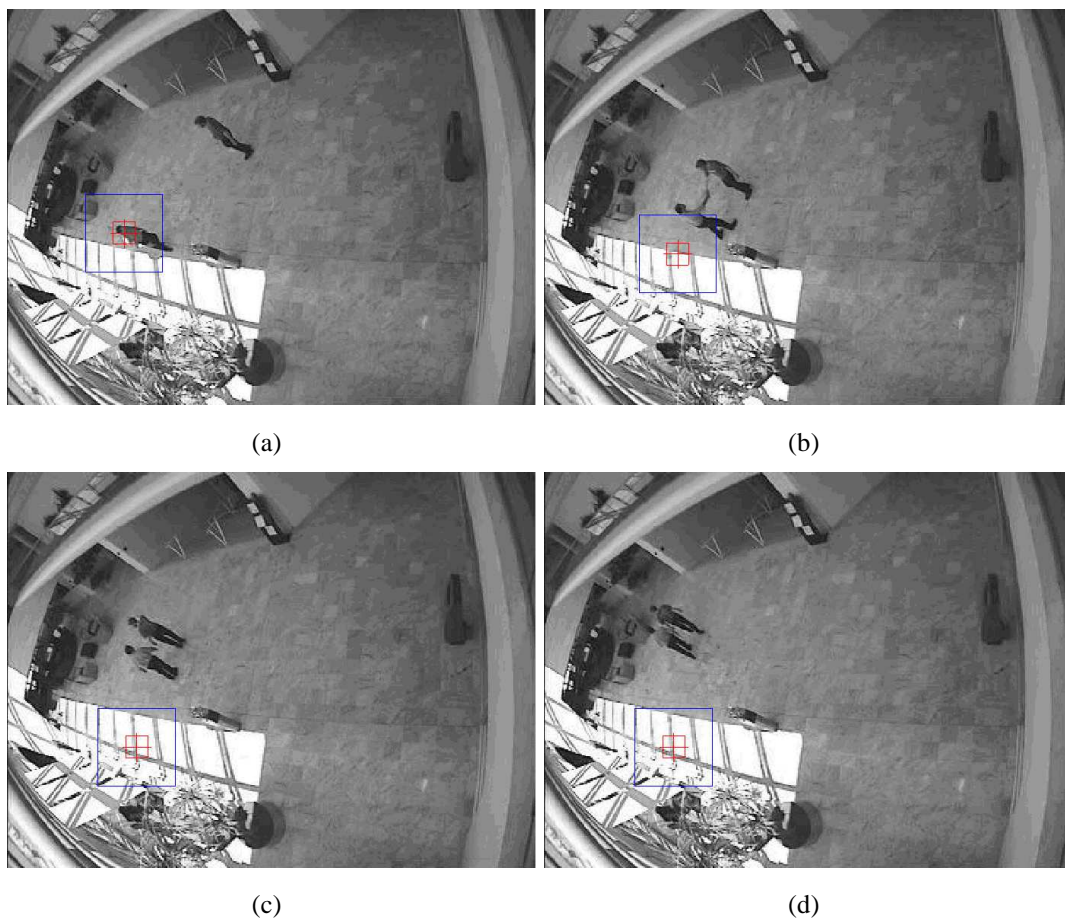


Figura 4.3: Resultado do rastreamento realizado pelo algoritmo Window-matching utilizando a função de Custo SSD para seqüência Encontro no Shopping.

Em seguida aplicou-se o algoritmo WM+K de rastreamento com filtragem estocástica. A figura 4.4 mostra alguns quadros do processo de rastreamento do indivíduo, nos quais a ROI do algoritmo WM+K aparece na cor amarela. A ROI do algoritmo WM, modificada pela filtragem estocástica, aparece na cor vermelha. Então, apesar do ambiente dispersivo, da ambigüidade existente, o rastreamento foi realizado com sucesso até o final da seqüência.

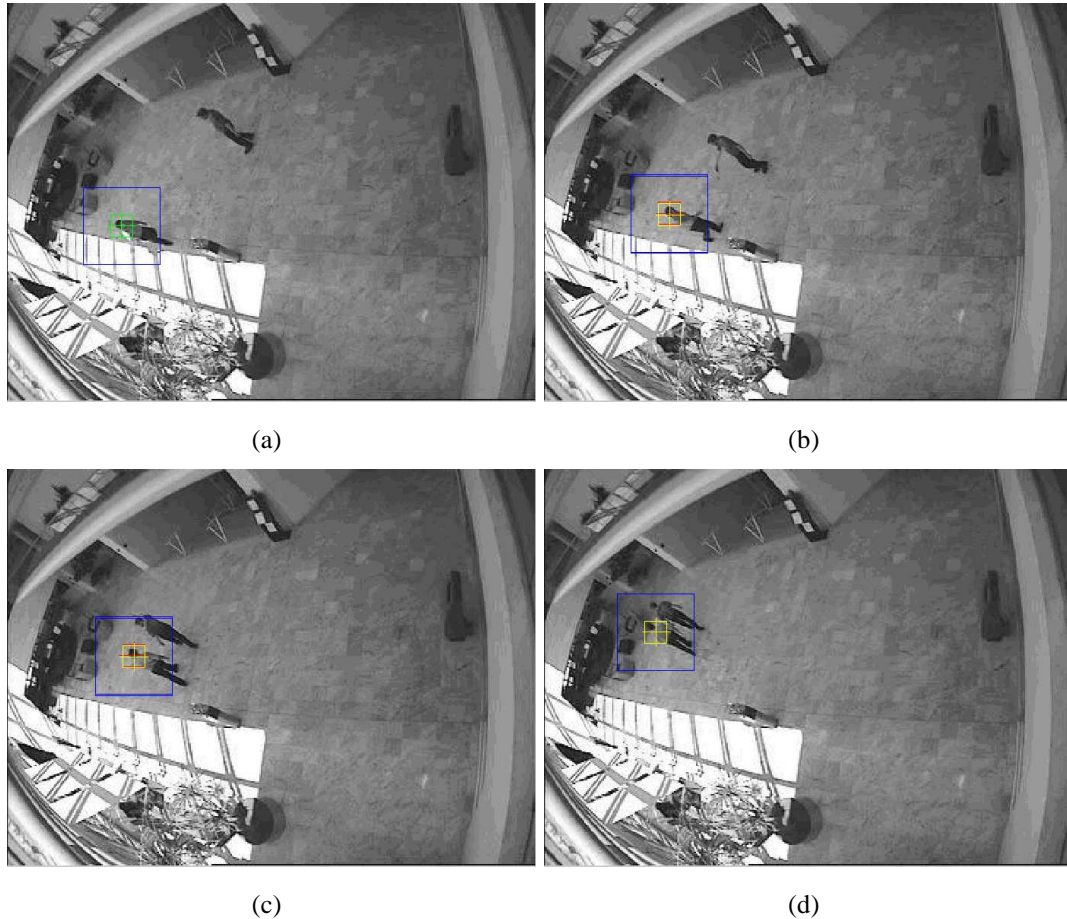


Figura 4.4: Resultado do rastreamento realizado pelo algoritmo WM+K para seqüência Encontro no Shopping.

A figura 4.5 mostra as posições horizontal e vertical do ponto da ROI representativo do indivíduo rastreado, como fornecidos pelo algoritmo WM+K. Para comparação, a figura mostra também as posições do algoritmo WM modificadas pela filtragem estocástica. Pode-se concluir então que a filtragem estocástica contribui para a "robustez" do processo de rastreamento.

Caso 2 - Encontro de duas pessoas em centro comercial⁴

A seqüência (Anexo B.5) mostra duas pessoas caminhando lado a lado ao longo de um corredor. Neste caso existe baixa distorção fométrica, mas acontece uma oclusão parcial causada pela passagem de uma terceira pessoa. A oclusão é um dos problemas que fazem o problema de rastreamento mais complexo e muitas vezes insolúvel. A oclusão é tratada em

⁴Seqüências de imagens obtidas em <http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/>

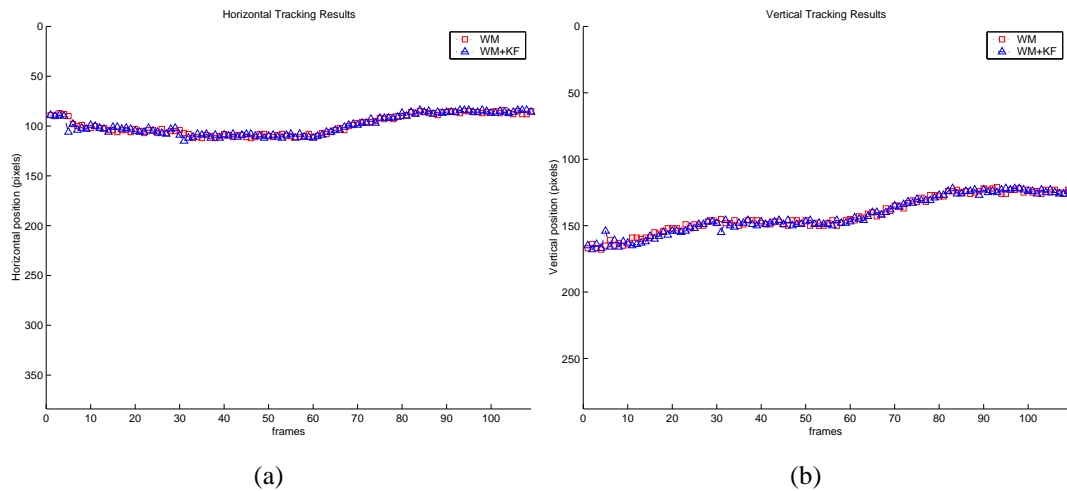


Figura 4.5: Posições horizontal(a) e vertical(b) descritas pelo rastreamento WM+K para seqüência Encontro no Shopping. Legenda: \square WM e em \triangle WM+K

diversos trabalhos [22], [76], [77], [78] e [79], mostrando soluções para algumas situações particulares. As dimensões das imagens neste caso 2 são 348×288 pixels em formato *JPG* e a subregião de busca e ROI mais adequadas foram 40×40 e 10×10 pixels respectivamente.

Tal como no caso 1, primeiro aplicou-se o algoritmo WM e depois o WM+K. As figuras 4.6 e 4.7 mostram então os respectivos resultados na forma de alguns quadros do rastreamento. A ROI do algoritmo WM é demarcada na cor vermelha e a ROI do algoritmo WM+K na cor amarela.

Como se observa, o algoritmo WM não é capaz de fazer o rastreamento devido principalmente à oclusão parcial (quadro 3). Mas, o algoritmo WM+K manteve o rastreamento apesar da oclusão. Verifica-se então novamente a contribuição da filtragem estocástica à robustez do rastreamento. As posições horizontal e vertical do ponto representativo do indivíduo rastreado na aplicação do algoritmo WM+K são mostradas na figura 4.8.

Nota-se na Figura 4.8 que as principais causas das variações bruscas na trajetória de rastreamento são causadas pelo elevado grau de ambigüidade entre os elementos que compõe a cena. A ambigüidade proporciona ao algoritmo WM diversas regiões na imagem com valores de similaridade próximos, aumentando o efeito de deslocamento do aspecto da ROI. No decorrer do rastreamento, esse acúmulo de pequenos erros faz com que a ROI seja atualizada por outra, ocasionando no fracasso do rastreamento.

4.4.2 Rastreamento de Garrafa flutuando no mar

A seqüência (Anexo B.3) mostra uma garrafa flutuando no mar⁵. Esta seqüência tem particularidades no sentido que é um movimento aleatório não-rígido com variação de pro-

⁵Seqüências de imagens obtidas em <http://www.cs.bu.edu/groups/ivc/data/DynamicBackgrounds/ICCV2003/water/object7>

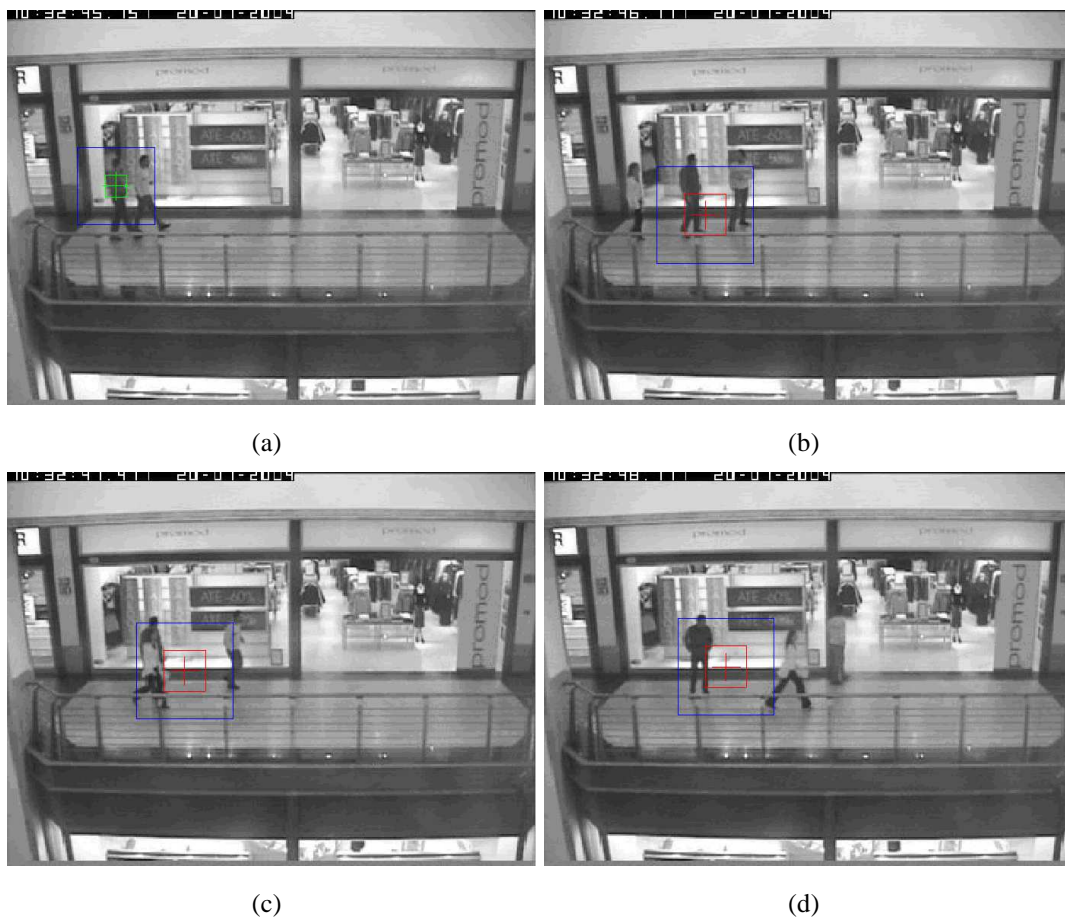


Figura 4.6: Resultado do rastreamento realizado pelo algoritmo WM para seqüência Caminhada no Shopping.

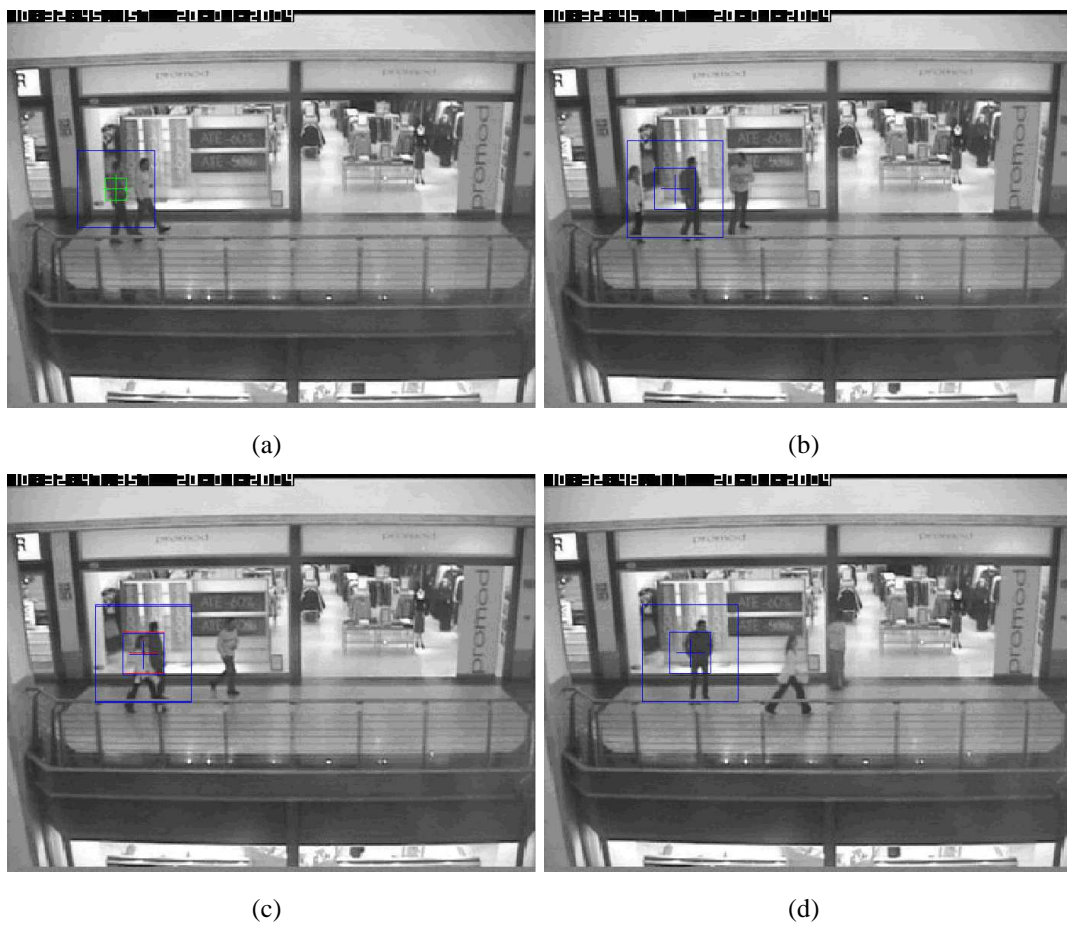


Figura 4.7: Resultado do rastreamento realizado pelo algoritmo WM+K para seqüência Caminhada no Shopping.

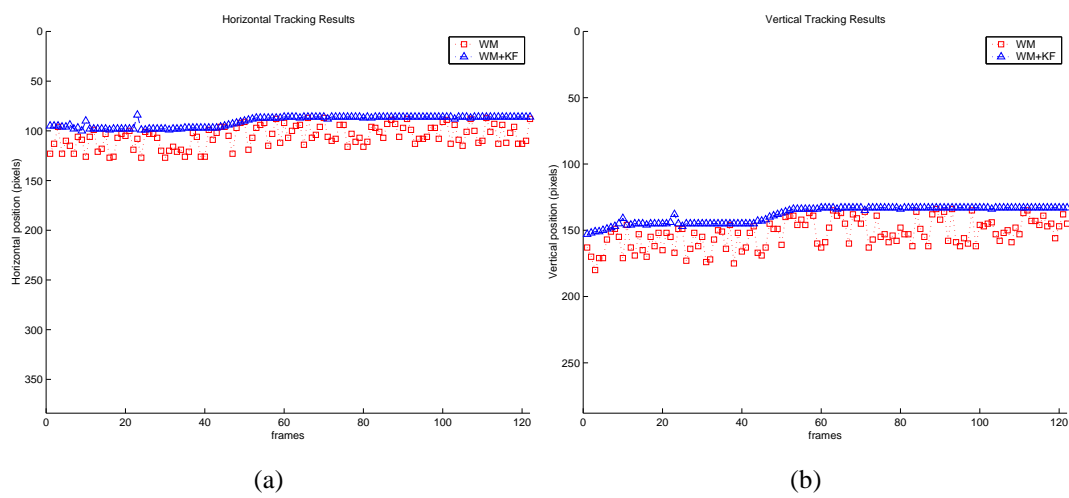


Figura 4.8: Posições horizontal(a) e vertical(b) descritas pelo rastreamento WM+K para seqüência Caminhada no Shopping. Legenda: \square WM e \triangle WM+K

fundidade, efeito de borramento (i. e. *blurring*), mudanças de escala e elevado grau de ambigüidade entre as regiões da imagem. Estas características demandam de algoritmo um maior grau de robustez para que o rastreamento seja feito com sucesso.

O problema, evidente neste caso, é o rastreamento da garrafa. Do mesmo modo aplica-se primeiro o algoritmo WM, cujos resultados são mostrados na Figura 4.9. Novamente, o algoritmo WM não é capaz de seguir a garrafa, principalmente devido à ambigüidade do ambiente. Já o algoritmo WM+K consegue fazer o rastreamento como observado na Figura 4.10.

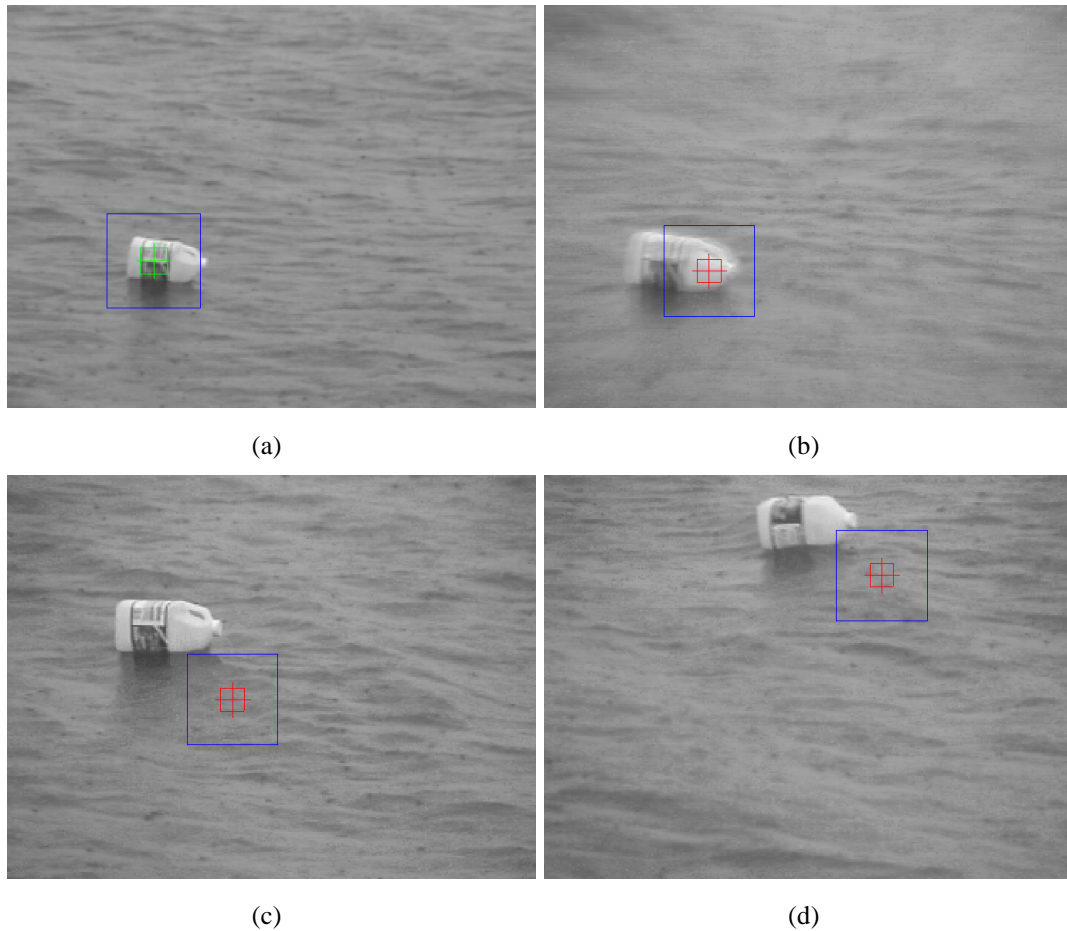


Figura 4.9: Resultado do rastreamento realizado pelo algoritmo WM para seqüência objeto flutuando no mar.

A Figura 4.11 mostra as posições horizontal e vertical do ponto de referência da garrafa como resultado do rastreamento pelo algoritmo WM+K. Devido à natureza aleatória dos movimentos da garrafa na superfície do mar, a contribuição da filtragem de Kalman foi fundamental. Observa-se uma pequena suavização da trajetória e a minimização do efeito deslocamento do aspectos da imagem (*feature drift*).

Para as seqüências aqui consideradas, as variações nas estimativas de rastreamento realizadas ocorreram, em sua grande maioria, quando os elementos da diagonal principal da matriz de covariância do erro (\mathbf{P}_k) mudava de sinal. Estas mudanças são causadas pelo ruído

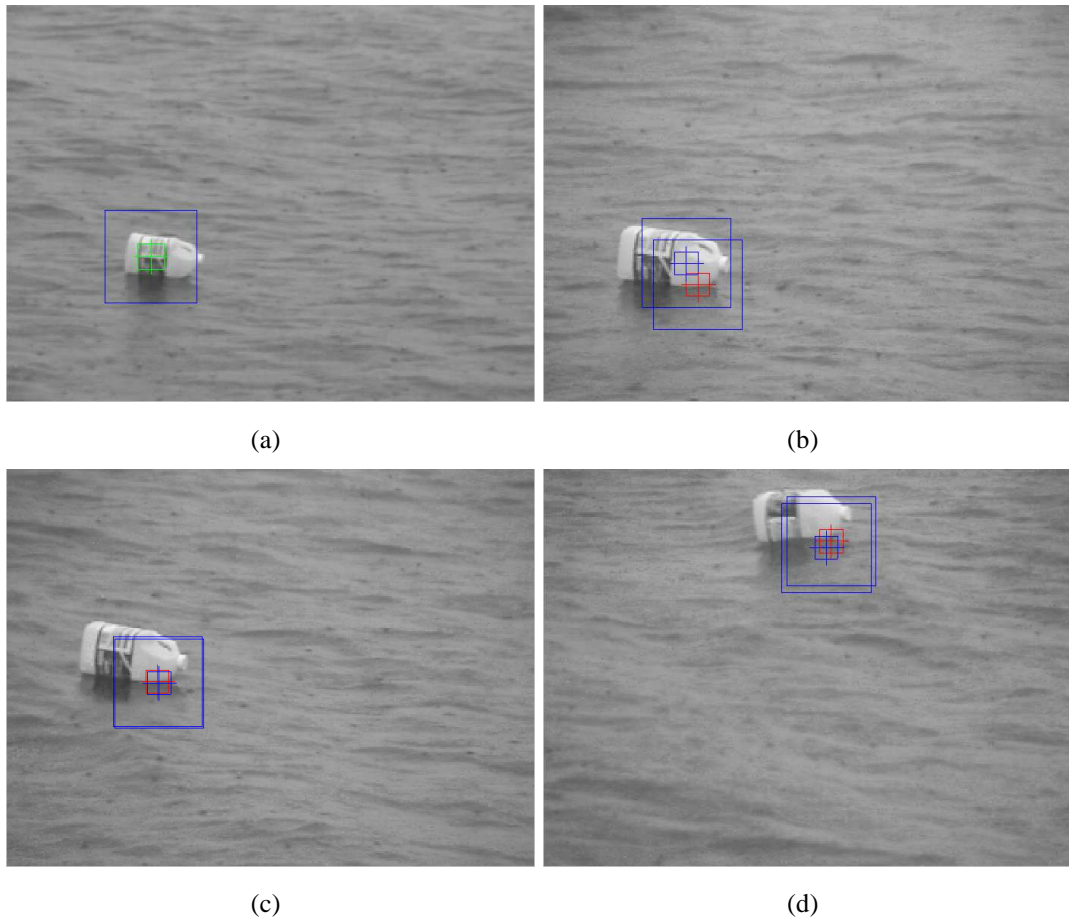


Figura 4.10: Resultado do rastreamento realizado pelo algoritmo WM+K para seqüência objeto flutuando no mar.

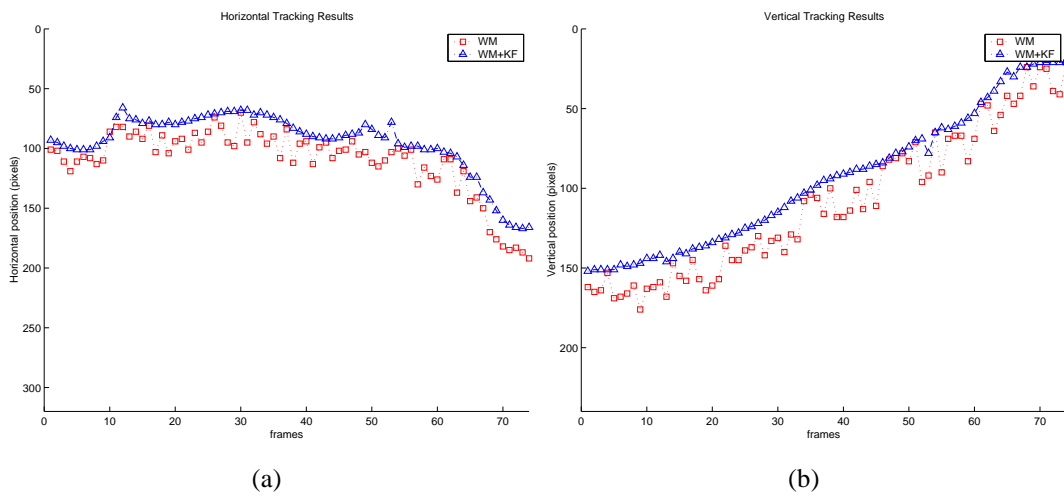


Figura 4.11: Posições horizontal(a) e vertical(b) descritas pelo rastreamento WM+K para seqüência objeto flutuando no mar. Legenda: \square WM e \triangle WM+K

de medições introduzidos pelo Filtro de Kalman. A proposta do algoritmo WM+K era de realizar o rastreamento visual, com melhores resultados que o algoritmo WM e oferecendo maior robustez ao processo de rastreamento, minimizando os efeitos das principais adversidades que envolvem este tipo de processo. Contudo, ainda há espaço para melhorias, estas serão apresentadas como propostas no próximo capítulo persegue-se por exemplo melhorar ainda mais o tratamento do efeito de *drift*, das variações de trajetórias indesejadas e das metodologias de decisões do Filtro de Kalman, correlacionando não só as medidas atuais de posição do algoritmo WM e WM+K, mas incorporando informações das posições passadas.

5 OTIMIZAÇÃO SEGUNDO AGRUPAMENTO DE PARTÍCULAS APLICADA AO RASTREAMENTO VISUAL

O algoritmo de correspondência de janelas (WM) mostrou-se satisfatório para alguns casos de rastreamento, mas em casos mais complexos, com ambiguidades e alguma oclusão, o algoritmo falhou. A introdução da filtragem estocástica no algoritmo WM contribuiu na robustez do processo de rastreamento e esses casos foram resolvidos pelo algoritmo de correspondência de janelas acrescido de filtragem estocástica (WM+K). Mas, existem casos mais complexos ainda, como situações onde existem: a) variações bruscas de trajetória produzidas pelo efeito de deriva dos aspectos da imagem (*feature drift*); e b) oclusões parciais e/ou trajetórias incompletas no deslocamento da região de interesse (ROI) em torno do objeto. Em sequências de imagens que apresentavam estas complexidades, o algoritmo WM+K nem sempre apresentou resultados satisfatórios. Observou-se que, na situação (a), o problema no desempenho do algoritmo era a mudança de sinal no ganho do filtro de Kalman causado pelas variações bruscas de trajetória. Já a situação (b) é um desafio constante na solução do problema de rastreamento.

Na investigação de uma melhora no algoritmo para lidar com estas situações estudou-se primeiro a utilização de formas estendidas do filtro de Kalman. Uma abordagem heurística de simples implementação e que trouxe resultados foi a aplicação da otimização segundo agrupamento de partículas (Particle Swarm Optimization - PSO) [80]. Este capítulo trata então da inserção desta otimização no rastreamento visual de objetos. Na seção 5.1 um breve histórico desta otimização heurística é relatado. A seção 5.2 segue com a descrição matemática da otimização baseada no comportamento de um grupo de partículas. A seção 5.3 mostra a introdução da restrição de Lipschitz na otimização, isso decorre dos problemas da implementação da PSO para o rastreamento.

5.1 UM BREVE HISTÓRICO

A otimização segundo agrupamento de partículas (PSO) é um método de otimização heurística de funções não-lineares e foi apresentado por Kennedy e Eberhard [80] em 1995. O método foi descoberto através da simulação de um modelo social simplificado do comportamento social de bandos de pássaros, cardumes de peixes, enxames de insetos, etc. Daí derivou para um método de otimização, onde pássaros, peixes ou insetos constituem as partículas. Perseguindo um objetivo estas partículas devem os seus movimentos à inércia, à experiência adquirida e à experiência das demais partículas. Ou seja, o movimento da partícula resulta da ponderação de três componentes: inércia, memória e cooperação. O método PSO está de alguma forma relacionado com a computação bio-inspirada.

Inicialmente sem nenhuma orientação prévia, essas partículas se movimentam até que uma delas consiga encontrar a melhor posição (melhor estado), em função de um objetivo pré-definido (função de custo), que acaba por atrair as partículas mais próximas, assim desenvolvendo o princípio da inteligência social. De forma semelhante aos outros métodos de otimização natural, o PSO trabalha com populações de soluções, que em uma mesma iteração, vários pontos do espaço de busca são avaliados através da função de custo. Apesar de demandar um custo computacional significativo, o PSO possui uma maior capacidade de percorrer o espaço de busca, aumentando a probabilidade de obtenção de ótimos globais [81].

Trabalhos relevantes, que empregam a técnica de otimização por agrupamento de partículas, para a realização do rastreamento visual e/ou processamento de imagens são encontrados em [82], [83], [84] e [85]. Em [83] o algoritmo PSO é utilizado para detecção de pessoas em imagens de infravermelho, em que cada partícula foi tratada como um detector com escala específica. Todas as partículas mapeiam a imagem para encontrar pessoas, nas quais o tamanho se adapta a partícula. Akbari *et al.* ([82]) desenvolveu uma estrutura que emprega o PSO em conjunto com metodologias de filtragem estocástica. Cada objeto a ser rastreado é dividido em blocos e cada bloco é representado por uma partícula. As partículas são guiadas pelo filtro estocástico(ex.: um Filtro de Kalman) para realizar o rastreamento do objeto. Em [84] foi criada uma estrutura baseada no algoritmo PSO, a qual foi empregada diretamente nos pixels da imagem e onde o objeto rastreado era detectado pela interação entre as partículas.

5.2 O MÉTODO PSO CLÁSSICO

O método consiste na otimização de uma função objetivo onde cada possível solução será uma partícula. Inicialmente todas as partículas estão posicionadas aleatoriamente no espaço de busca, no qual espera-se encontrar a solução ótima. O movimento de cada partícula é considerado ser a soma vetorial de três contribuições distintas devidas : a) à inércia da partícula; b) à memória da partícula (melhor solução local); e c) à cooperação das outras partículas (melhor solução global). Se na iteração k a posição da i -ésima partícula é x_i , sua posição na iteração seguinte $k + 1$ será dada por

$$x_{k+1}^i = x_k^i + v_{k+1}^i \Delta t \quad (5.1)$$

onde v_{k+1} é o vetor velocidade da partícula i e Δt é o intervalo de tempo entre iterações. A mudança de posição da partícula depende do vetor velocidade que, considera-se determinado por uma ponderação das três contribuições segundo

$$v_{k+1}^i = W_k v_k^i + c_1 r_1 \frac{(p^i - x_i^k)}{\Delta t} + c_2 r_2 \frac{(p_k^s - x_i^k)}{\Delta t} \quad (5.2)$$

onde: p_i : melhor posição encontrada pela partícula i (solução local); p_k^s : melhor posição encontrada pelo grupo na iteração k (solução global); r_1 e r_2 : números aleatórios no intervalo $[0, 1]$; c_1 e c_2 : parâmetros indicadores da confiança concedida às melhores soluções local e global respectivamente; W_k : vetor de ponderação relacionado com a inércia das partículas na iteração k , este vetor é obtido a partir de

$$W_k = W_0 - \frac{k(W_0 - W_{k-1})}{K} \quad (5.3)$$

sendo:

k : número máximo de iterações; W_0 e W_{k-1} : valores inicial e anterior dos vetores de ponderação de inércia.

Valores baixos dos vetores de inércia favorecem uma busca local e valores altos uma busca global [86]. Os valores do vetor de inércia (W) e dos parâmetros de confiança (c_1 e c_2) devem ser ajustados de acordo com o problema a ser resolvido. O método PSO foi então implementado e foram realizados testes onde observou-se algumas vezes problemas de convergência do algoritmo ou seja uma solução ótima não era encontrada. Uma restrição foi então inserida, como tratado a seguir.

5.3 MÉTODO PSO COM A RESTRIÇÃO DE LIPSCHITZ

A procura de soluções no rastreamento visual foi abordada como um problema de otimização, para o qual aplicou-se o método PSO, onde as posições das partículas correspondem às possíveis soluções. A implementação seguiu as equações 5.1 e 5.2 e nos testes observaram-se problemas de convergência do algoritmo. Na aplicação do método PSO na otimização do desempenho de sistemas dinâmicos, principalmente para funções de custo quadráticas, recomenda-se a introdução da restrição de Lipschitz [87]. Esta restrição limita a velocidade da partícula v_k sendo representada por

$$|v_{k+1}^i| \leq V_{max} \quad \text{ou} \quad |x_{k+1}^i - x_k^i| \leq V_{max} \cdot \Delta t \quad (5.4)$$

A otimização PSO com a restrição (Equação 5.4) garante a convergência do algoritmo segundo [87], ou seja a obtenção de uma solução entre as posições locais e globais das partículas. Introduziu-se então esta busca heurística no rastreamento visual por correspondência de janelas com filtragem estocástica.

5.4 MÉTODO DE RASTREAMENTO POR CORRESPONDÊNCIA DE JANELAS COM FILTRAGEM ESTOCÁSTICA E BUSCA PSO

Para aumentar a capacidade de rastreamento visual, o algoritmo desenvolvido no capítulo 4, de correspondência de janelas com filtragem de Kalman (WM+K), foi então modificado para a inserção da otimização segundo agrupamento de partículas (PSO). O novo algoritmo foi chamado de algoritmo WM+K+PSO. Na execução do algoritmo WM+K, o estágio de correspondência de janelas (WM) fornece os vetores de posição segundo o grau de similaridade, o estágio de filtragem de Kalman (K) processa estes vetores e gera estimativas com indicação da tolerância do erro estimação. No caso do estágio WM não fornecer um vetor de posição válido, a estimativa deste vetor gerada pelo filtro de Kalman tomará seu lugar. Comprovou-se que assim aumentava-se a robustez do algoritmo.

Durante testes observou-se que em casos severos de deriva de aspectos da imagem (*feature drift*) o algoritmo WM+K não conseguia fazer o rastreamento. Do estudo destes casos concluiu-se que a deriva pronunciada causava variações na matriz de covariância \mathbf{P} e por consequência variações indesejadas no ganho do filtro de Kalman, o que corrompia as estimativas. Isto é corroborado em [88]. Para resolver o problema de rastreamento nestes casos introduziu-se a busca heurística de soluções baseada na otimização segundo agrupamento de partículas com a restrição de *Lipschitz*.

Outras estratégias foram adotadas visando a diminuição do custo computacional elevado do algoritmo e conseqüentemente o tempo de execução do algoritmo. Entre elas, pode-se citar o uso de *sub-regiões* da região de interesse. A escolha da região de interesse é definida de acordo com os objetos em que se deseja rastrear. Outros fatores que influenciam na escolha da região são: o grau de ambigüidade, variações súbitas de iluminação, direção e profundidade. Precisamente, a adição da filtragem de Kalman ao algoritmo ajudou a diminuir consideravelmente os efeitos causados por estes fatores.

5.4.1 Desenvolvimento de um Algoritmo de Rastreamento por Correspondência de Janelas com Filtragem de Kalman e Otimização segundo Agrupamento de Partículas - Algoritmo WM+K+PSO

Implementou-se então na forma de algoritmo a otimização segundo agrupamento de partículas (PSO) com restrição de Lipschitz, sua estrutura é mostrada na Figura 5.1.

A Figura 5.2 mostra então a estrutura do algoritmo WM+K+PSO, que integra a otimização PSO no algoritmo de rastreamento WM+K.

O estágio de correspondência de janelas fornece um vetor medido de posições e velocidades $[x_{WM} \ v_{WM}]$ e o estágio de filtragem de Kalman, um outro vetor estimado $[x_K \ v_K]$. Inicialmente procedeu-se à otimização de ambas, posições e velocidades. Observou-se que o efeito da deriva de aspectos da imagem (*feature drift*) era magnificado, prejudicando a

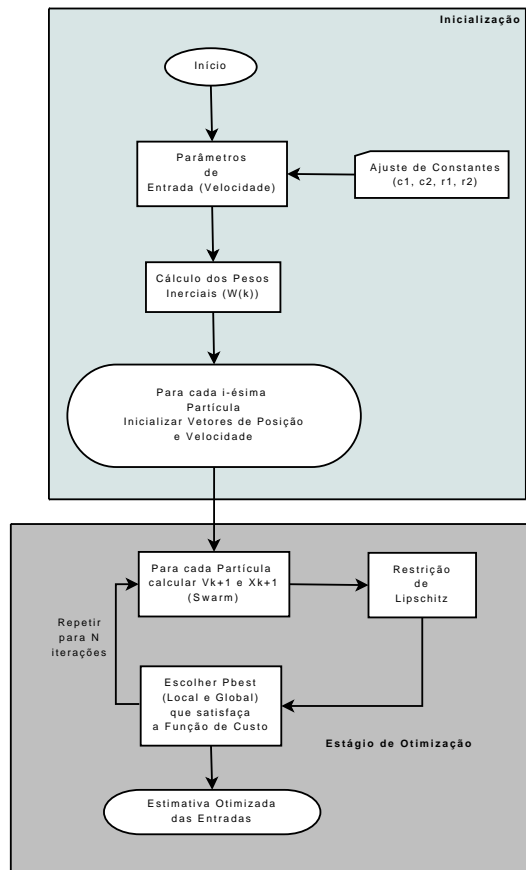


Figura 5.1: Algoritmo PSO Clássico com restrição de Lipschitz

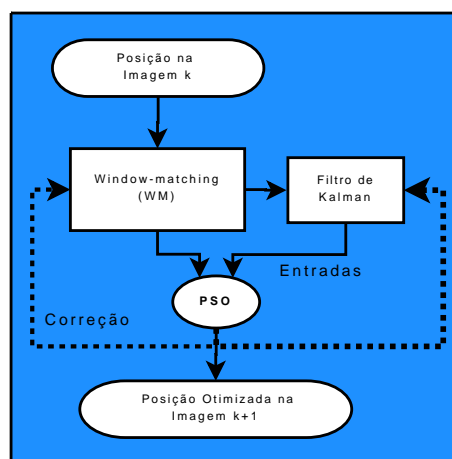


Figura 5.2: Metodologia de Acoplamento dos métodos PSO e Window-Matching com Filtro de Kalman

convergência do algoritmo. Sabendo a maior importância da velocidade na convergência do filtro de Kalman nos sistemas dinâmicos [88], optou-se pela otimização unicamente da velocidade. Isto é coerente com a restrição de *Lipschitz* que restringe apenas as velocidades.

Definiu-se então uma função de custo V_{erro} para cada instante de tempo T . Ela servirá para a correção das velocidades horizontal e vertical v_{WM} e v_K . A função de custo foi definida então como

$$\min V_{erro} = \frac{1}{T} \sum [v_{WM} - v_K]^2. \quad (5.5)$$

O acerto desta abordagem foi comprovada na convergência do algoritmo nos testes que se seguiram.

5.5 APLICAÇÕES DO ALGORITMO WM+K+PSO NO RASTREAMENTO VISUAL

Implementado o algoritmo seguindo a Figura 5.2, também no ambiente Matlab escolheram-se 4 seqüências de teste:

- a) Encontro e caminhada de pessoas em local público;
- b) Garrafa flutuando no mar;
- c) Humano com movimentos determinísticos;
- d) Humano com movimentos aleatórios.

As duas primeiras foram utilizadas no Capítulo 4 e só o rastreamento pelo algoritmo WM+K conseguiu resultados satisfatórios. As duas seguintes apresentam desafios como oclusões e movimentos aleatórios respectivamente, nas quais o algoritmo WM+K mostrou dificuldades no rastreamento. As soluções dos algoritmos WM supervisionado, WM+K e WM+K+PSO são apresentados nas cores vermelha, amarela e verde respectivamente.

5.5.1 Rastreamento de Indivíduos em Sequência de Circulação de Pessoas em Locais Públicos

As informações sobre esta seqüência foram dadas na subseção 4.4.1. Os resultados de aplicação dos algoritmos WM e WM+K foram mostrados nas Figuras 4.6, 4.7 e 4.8. A seqüência como foi dito apresenta oclusões parciais mas baixa ambiguidade. Na aplicação do algoritmo WM+K+PSO configurou-se também uma subregião de busca de 40×40 pixels e uma ROI de 10×10 pixels. O estágio de otimização PSO foi configurado com 100 partículas com as informações das posições e velocidades nas direções horizontal e vertical fornecidas pelo estágio WM+K. Os parâmetros indicadores da confiança c_1 e c_2 concedidas às melhores soluções local e global foram ajustadas ambas no número 2 e os números aleatórios r_1 e r_2 ajustados ambos em 1. Já os coeficientes do vetor inercial foram ajustados na faixa de 0, 1 a

0, 9. Finalmente a restrição de *Lipschitz* na velocidade foi ajustada em 100.

A Figura 5.3 mostra alguns quadros do processo de rastreamento de um indivíduo pelo algoritmo WM+K+PSO, mostrando as saídas dos estágios WM e K. As trajetórias, em termos das posições na horizontal e na vertical aparecem nas Figuras 5.4 - (a) e (b) respectivamente. A comparação com os resultados do algoritmo WM+K (Figura 4.8) mostra que não existem oscilações indesejáveis na trajetória obtida do rastreamento, claramente uma vantagem do algoritmo WM+K+PSO.

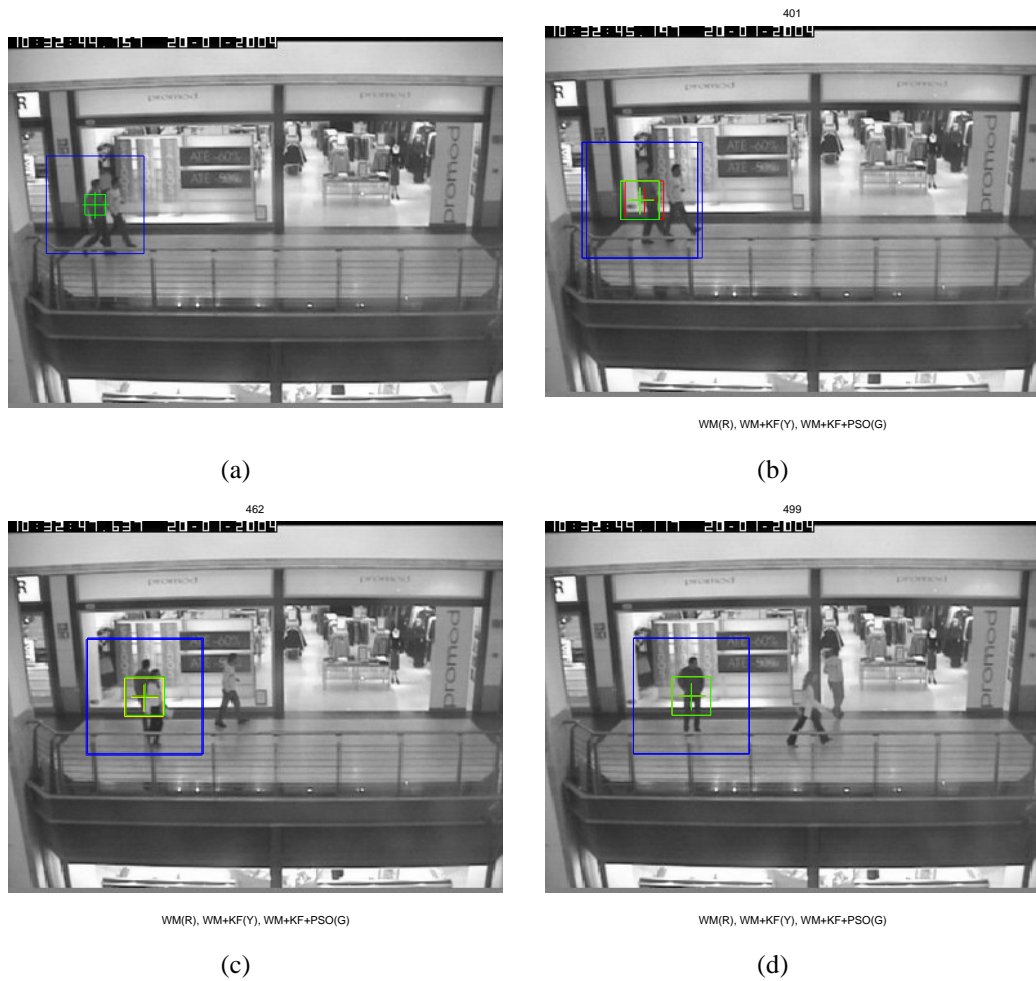


Figura 5.3: Resultado do rastreamento realizado pelo algoritmo WM+K+PSO para seqüência Caminhada no Shopping.

5.5.2 Rastreamento de Garrafa flutuando no mar

As informações sobre esta seqüência foram dadas na subseção 4.4.2. Os resultados de aplicação dos algoritmos WM e WM+K foram mostrados nas Figuras 4.9, 4.10 e 4.11. Embora a garrafa seja um objeto rígido, esta realiza movimento equivalentes a objetos deformáveis. Mais ainda o fundo (mar) eleva o grau de ambigüidade das cenas. O estágio de otimização PSO foi configurado do mesmo modo que na seqüência anterior (subseção 5.5.1).

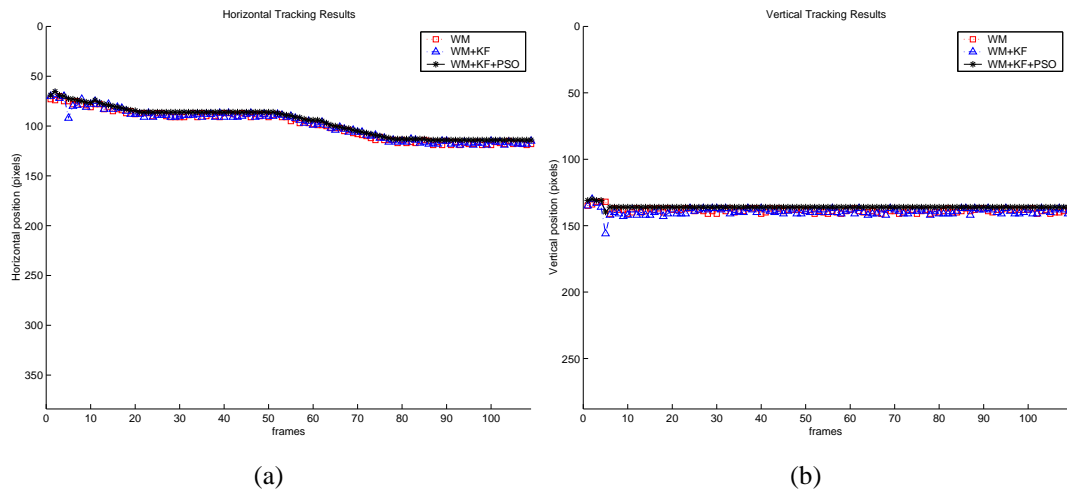


Figura 5.4: Posições horizontal(a) e vertical(b) descritas pelo rastreamento WM, WM+K e WM+K+PSO para seqüência Caminhada no Shopping. Legenda: \square WM, \triangle WM+K e $— * —$ WM+K+PSO.

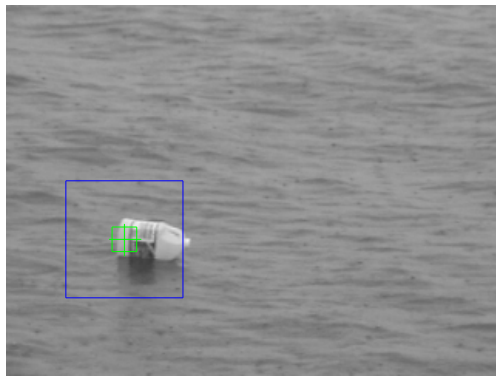
A Figura 5.5 mostra alguns quadros do processo de rastreamento da garrafa pelo algoritmo WM+K+PSO, mostrando as saídas dos estágios WM e K. As trajetórias, em termos das posições na horizontal e na vertical aparecem nas Figuras 5.6 - (a) e (b) respectivamente. A comparação com os resultados do algoritmo WM+K (Figura 4.10) que o efeito da deriva foram praticamente eliminados, não aparecem as oscilações e a trajetória obtida do rastreamento é suave.

5.5.3 Rastreamento de face humana com ocorrência de oclusões parciais

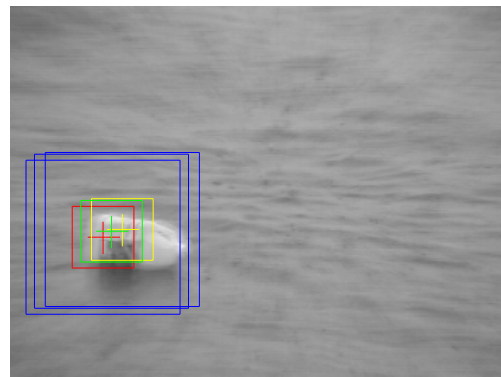
Nesta seqüência¹ (Anexo B.6) aparece um indivíduo sentado fazendo movimentos limitados da sua face e movimentos mais amplos com sua mão com os quais sua face fica parcialmente coberta. Trata-se então de rastrear a face do indivíduo na presença destas oclusões parciais e do efeito de "borramento" causado pela baixa taxa de aquisição das imagens. Estas características apresentaram um desafio para o teste de robustez do algoritmo WM+K+PSO. Os parâmetros de configuração do algoritmo foram os mesmos da seqüência anterior, a menos das dimensões da subregião de busca e da ROI, que foram 120×120 e 80×80 respectivamente. Como já discutido anteriormente, as dimensões destas janelas devem ser ajustadas de acordo com o problema tratado. O aumento de tamanho destas janelas trouxe, claro, um aumento da carga computacional. Sobre isto o Anexo 5.6 apresenta um quadro comparativo sobre a carga computacional de cada algoritmo desenvolvido.

A Figura 5.7 mostra alguns quadros do processo de rastreamento da face do indivíduo, mostrando ainda a saída dos outros estágios. As Figuras 5.8 - (a) e (b) mostra então as componentes horizontal e vertical das trajetórias rastreadas. Novamente observa-se o "aplai-

¹Seqüência obtida em <http://www.cs.toronto.edu/~dross/ivt/>

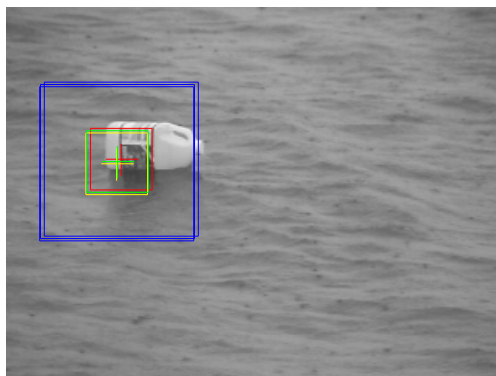


(a)



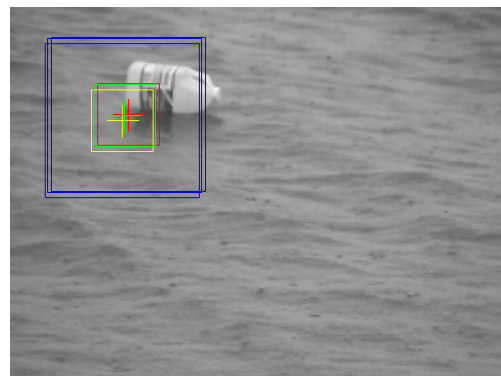
WM(R), WM+KF(Y), WM+KF+PSO(G)

(b)



WM(R), WM+KF(Y), WM+KF+PSO(G)

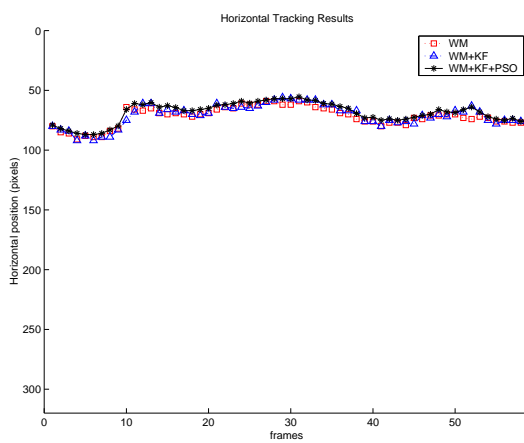
(c)



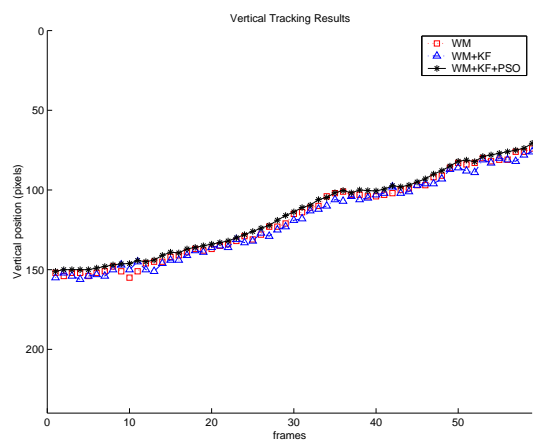
WM(R), WM+KF(Y), WM+KF+PSO(G)

(d)

Figura 5.5: Resultado do rastreamento realizado pelo algoritmo WM+K+PSO para seqüência objeto flutuando no mar.



(a)



(b)

Figura 5.6: Posições horizontal(a) e vertical(b) descritas pelo rastreamento WM, WM+K e WM+K+PSO para seqüência objeto flutuando no mar. Legenda: \square WM, em \triangle WM+K e em $- * -$ WM+K+PSO.

namento" pelo estágio PSO dos "saltos" produzidos pelo estágio de filtragem de Kalman.

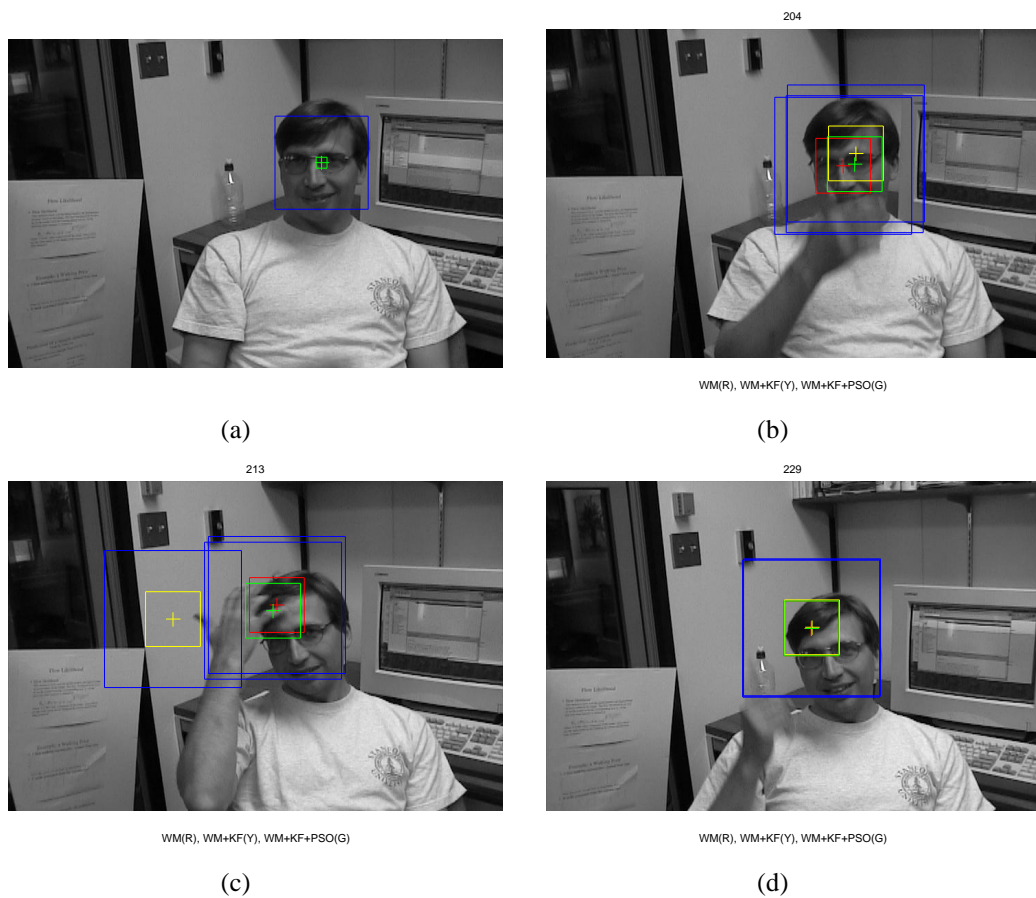


Figura 5.7: Resultado do rastreamento realizado pelo algoritmo WM+K+PSO para seqüência Face humana com oclusões parciais.

5.5.4 Rastreamento de um objeto movimentado de forma aleatória por um indivíduo

Nesta seqüência (Anexo B.7) aparece um indivíduo em um ambiente dispersivo movimentando um objeto rígido de forma aleatória². Trata-se então de rastrear o objeto em condições de variações não-uniformes de iluminação, de variações bruscas pela também baixa taxa de captura de imagens e pela presença de outros objetos nas imagens. A configuração do algoritmo foi a mesma do caso anterior.

A Figura 5.9 mostra então alguns quadros do processo de rastreamento do objeto, mostrando também a saída dos outros estágios. As Figuras 5.10 - (a) e (b) mostram as componentes horizontal e vertical das trajetórias rastreadas. Novamente observa-se a suavização das trajetórias pelo estágio PSO e sua contribuição evitando a "perda" do rastreamento do objeto, que aconteceu com os algoritmos WM e WM+K individualmente aplicados. Deste modo a otimização PSO representa uma contribuição para a solução do problema de rastreamento.

²Seqüência extraída de <http://esm.gforge.inria.fr/ESMdownloads.html>

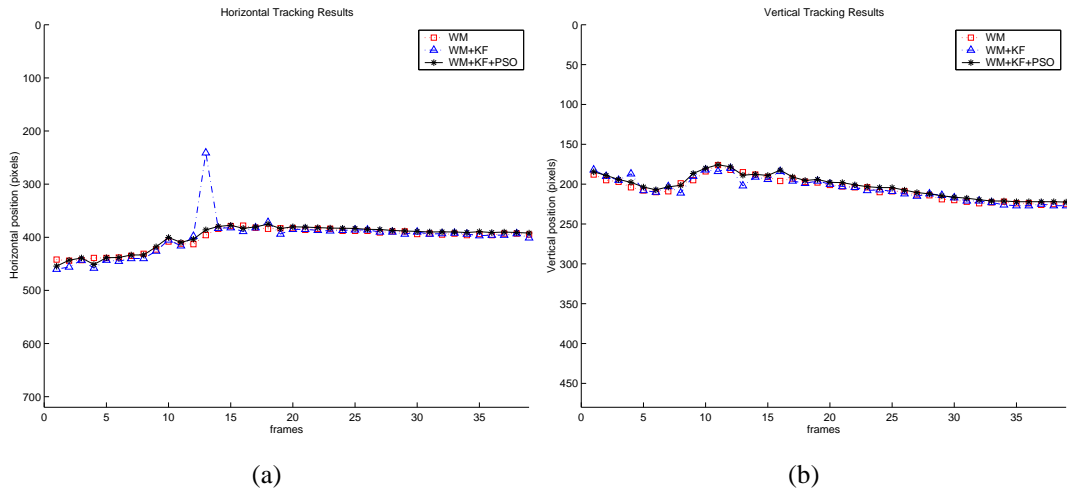


Figura 5.8: Posições horizontal(a) e vertical(b) descritas pelo rastreamento WM, WM+K e WM+K+PSO para seqüência rastreamento de face humana com oclusões parciais. Legenda: \square WM, em \triangle WM+K e em $- * -$ WM+K+PSO.

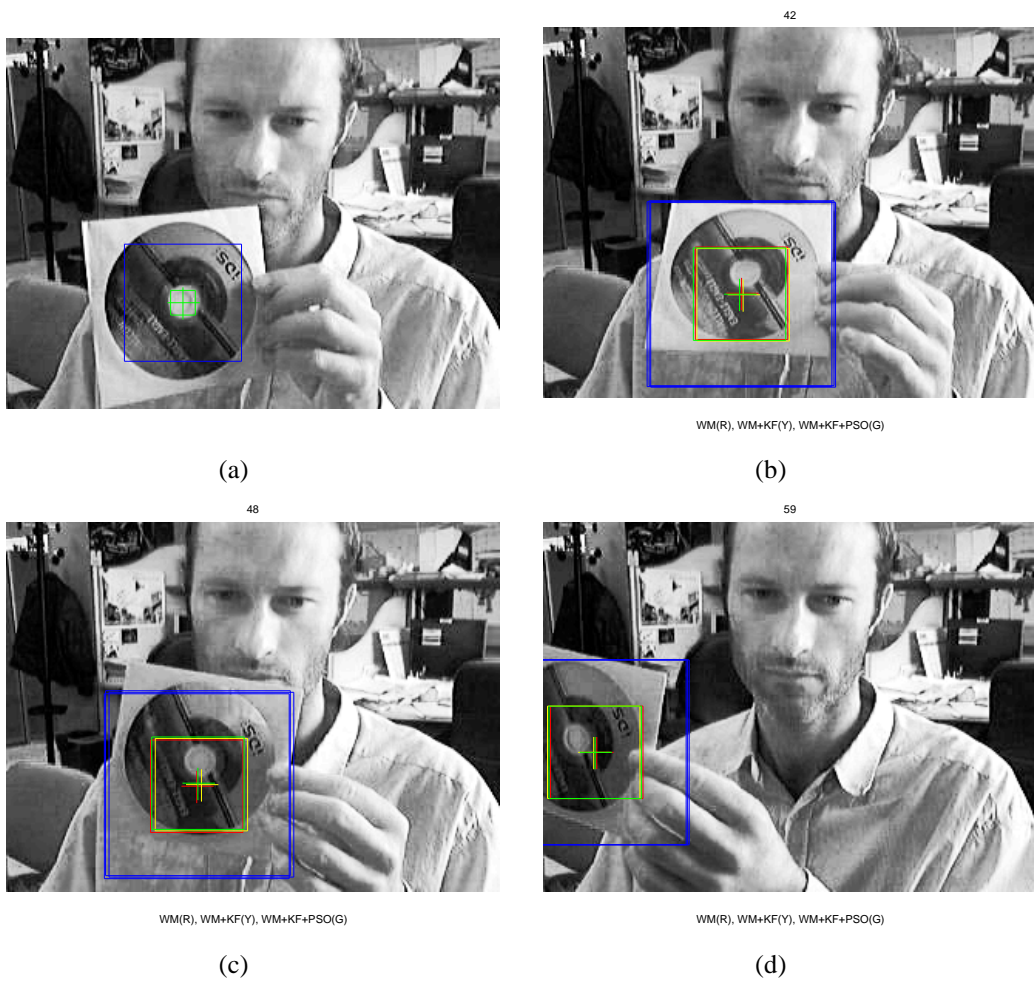


Figura 5.9: Resultado do rastreamento realizado pelo algoritmo WM+K+PSO para seqüência objeto com movimentos aleatórios. Legenda: Marcação em vermelho - WM, em amarelo - WM+K e verde - WM+K+PSO

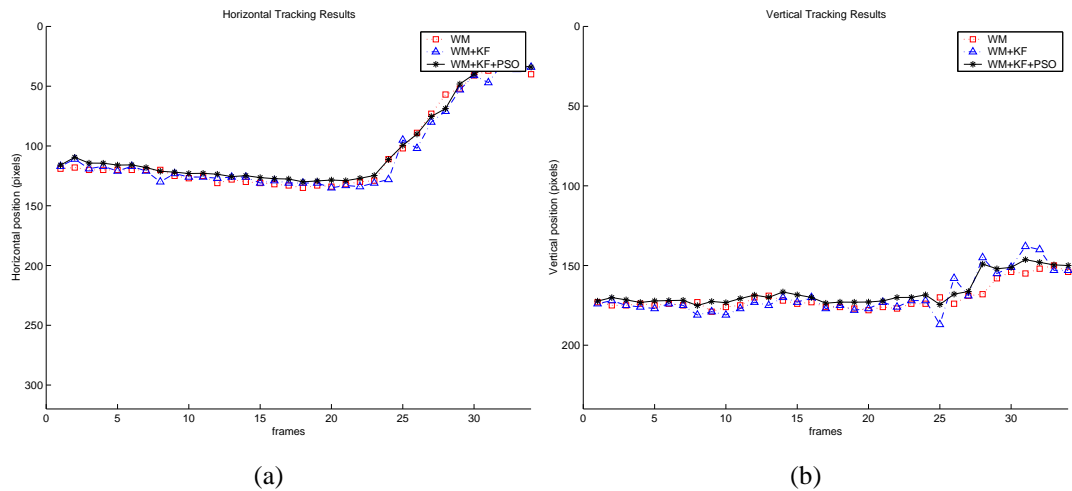


Figura 5.10: Posições horizontal(a) e vertical(b) descritas pelo rastreamento WM, WM+K e WM+K+PSO para seqüência rastreamento objeto com movimentos aleatórios. Legenda: \square WM, em \triangle WM+K e em $- * -$ WM+K+PSO.

5.6 TABELA COMPARATIVA DO CUSTO COMPUTACIONAL DOS ALGORITMOS DESENVOLVIDOS

Este tipo de abordagem acarreta problemas referentes à obtenção da medida do tempo por dois motivos: O primeiro deles é em relação ao tipo de hardware utilizado nas medições, o qual deve ser o mesmo utilizado em todas as metodologias. O segundo motivo é referente ao ambiente de desenvolvimento e ao sistema operacional utilizado, que principalmente em aplicações em tempo real, requerem controle rigoroso quanto à garantia de realização temporal da tarefa, independente dos outros processos concorrentes.

Ao utilizar como parâmetro o número de operações em pontos flutuantes (denominados *flops* no ambiente computacional Matlab[®]), consegue-se eliminar o problema referente ao hardware de processamento utilizado. Todas as implementações foram desenvolvidas em processadores 64 bits (AMD Turion 64-X2) com ponto flutuante e Sistema Operacional baseado na família Microsoft Windows Vista[®].

Legenda:

B.2 - Seqüência cruzamento de trânsito em grandes centros urbanos;

B.1 - Seqüência Jogo de Tênis de Mesa;

B.3 - Seqüência Objeto flutuando no mar;

B.4/B.5 - Seqüência Encontro de pessoas em interiores de edifícios em locais públicos;

B.6 - Seqüência de face humana com oclusões parciais;

B.7 - Seqüência de objeto com movimentos aleatórios;

Nome da Sequência	Num. Frames	Algoritmo	Flops/Frame(média)	Flops Total Frames	Total Geral
B.1	48	WM	308.135	14.790.512	16.973.988
		WM+K	1.299	62.392	
		WM+K+PSO	44.189	2.121.084	
B.2	49	WM	405.125	19.851.141	22.666.625
		WM+K	1.640	80.402	
		WM+K+PSO	55.818	2.735.082	
B.3	74	WM	383.439	28.374.504	32.626.452
		WM+K	1.640	121.416	
		WM+K+PSO	55.818	4.130.532	
B.4/B.5	409 / 799	WM	147.955 / 284.188	60.513.857 / 227.066.515	66.776.959 / 250.566.645
		WM+K	437 / 839	178.940 / 670.568	
		WM+K+PSO	14.875 / 28.572	6.084.162 / 22.829.562	
B.6	239	WM	263.362	62.943.522	65.184.416
		WM+K	267	63.992	
		WM+K+PSO	9.108	2.176.902	
B.7	59	WM	207.621	12.249.656	14.203.280
		WM+K	945	55.812	
		WM+K+PSO	32.166	1.897.812	

Figura 5.11: Custo computacional em operações com ponto flutuante dos algoritmos desenvolvidos para as seqüências utilizadas nos testes.

6 CONCLUSÕES

Neste trabalho abordou-se o problema do rastreamento visual de objetos móveis em uma seqüência de imagens. Discutiram-se os métodos de rastreamento e optou-se pelo método baseado na investigação da similaridade de regiões entre quadros consecutivos para determinar a trajetória do objeto. A diferença de similaridade é quantificada por uma função de custo e a sua minimização é o objetivo do método. As diferentes funções de custo foram estudadas e optou-se pela somatório das diferenças quadráticas (SSD).

Desenvolveu-se então um algoritmo de rastreamento baseado na similaridade de regiões que foi chamado de Algoritmo de Correspondência de Janelas (WM). Ele possui vantagens em termos de carga computacional se comparado com o método padrão, uma delas é a utilização de janelas de busca que diminuem o tempo de processamento. Outra vantagem é a inserção de um estágio de verificação da confiabilidade das medidas de similaridade. O algoritmo foi então aplicado a seqüência de teste, ele funcionou para algumas, mas falhou para outras.

A experiência obtida levou à introdução da filtragem estocástica para compensar o efeito de perturbações (ambigüidades, oclusões parciais) que levaram o algoritmo a erro. Ou seja desejava-se aumentar a robustez do algoritmo, para o qual integrou-se um estágio de filtragem estocástica (filtro de Kalman). Embora existam métodos de rastreamento que utilizam o filtro de Kalman, aqui ele foi introduzido como 'supervisor' do estágio de correspondência de janelas (WM). Diversas implementações do filtro de Kalman, inclusive a forma estendida, foram estudadas e optou-se pelo filtro convencional para não carregar computacionalmente o algoritmo. Desenvolveu-se então o Algoritmo de Correspondência de Janelas com Filtragem Estocástica (WM+K). Nas aplicações este algoritmo foi capaz de realizar o rastreamento nas seqüências onde o algoritmo WM falhou, comprovando que contribuiu para a robustez do processo de rastreamento. Observou-se nas trajetórias obtidas oscilações e concluiu-se que elas eram devidas a mudanças de sinal nos elementos da matriz de covariância do filtro de Kalman. Estas mudanças afetam diretamente o ganho do filtro provocando as oscilações. Estas mudanças estão também relacionadas com a deriva dos aspectos da imagem (*feature drift*).

Para combater este problema estudou-se então uma busca heurística de soluções entre as medidas do estágio WM e as estimativas do estágio K. A busca heurística foi implementada aplicando a Otimização segundo Agrupamento de Partículas (PSO). Assim, desenvolveu-se o Algoritmo de Correspondência de Janelas com Filtragem Estocástica e Otimização segundo Agrupamento de Partículas (WM+K+PSO). O algoritmo foi aplicado a duas seqüências, para as quais o algoritmos anteriores mostraram deficiências, e obtiveram-se trajetórias suaves.

Estes algoritmos foram implementados no ambiente computacional Matlab[®] aprovei-

tando as vantagens propiciadas das facilidades de desenvolvimento. Este aspecto foi de fundamental importância, pois o interesse principal deste trabalho, está no desenvolvimento de novas metodologias que atendam requisitos para o rastreamento visual. Desta forma implementações para desenvolvimento em tempo real serão abordadas em trabalhos futuros.

O trabalho realizado permitiu também verificar, que seria pretencioso chegar a uma solução definitiva para um algoritmo de rastreamento. Ainda não existe na literatura este tipo de algoritmo. Os algoritmos desenvolvidos aqui são de fácil implementação e configuração e podem ser aplicados ao servocontrole visual.

Os algoritmos de rastreamento desenvolvidos, foram aplicados às seguintes seqüências:

- a) *Seqüência cruzamento de trânsito em grandes centros urbanos;*
- b) *Seqüência Jogo de Tênis de Mesa;*
- c) *Seqüência Encontro de pessoas em interiores de edifícios em locais públicos;*
- d) *Seqüência Objeto flutuando no mar;*
- e) *Seqüência de face humana com oclusões parciais; e*
- f) *Seqüência de objeto com movimentos aleatórios;*

A escolha destas seqüências aproxima a aplicação dos algoritmos desenvolvidos a situações reais, são situações que envolvem movimentos com rotações, translações e variações de profundidades diversas, bem como a ocorrência de oclusões parciais e rastreamento para movimentos aleatórios e não-rígidos.

A abordagem realizada nestas situações, permitiu melhorias no procedimento de rastreamento visual, devido ao fato que os algoritmos desenvolvidos são independentes do tipo de movimento realizado e permitindo também o rastreamento de objetos não rígidos e/ou independente à sua forma geométrica. Ainda não foi possível resolver o problema da oclusão total da região de interesse, tornando-se um objetivo de desenvolvimento em trabalhos futuros (ex. adição de sistemas com múltiplas câmeras).

Estes algoritmos também oferecem a flexibilidade de aplicações para situações, em que não há informação prévia sobre o objeto a ser rastreado. Esta característica permite que sua abrangência de aplicações seja expandida. Devido também às características de implementação serem simples, permite que seja desenvolvido independente da plataforma de desenvolvimento, permitindo ao desenvolvedor a escolha da linguagem e ambiente de desenvolvimento que melhor convier.

Uma evolução natural, para a continuação do desenvolvimento dos algoritmos aqui desenvolvidos, se faz na implementação em linguagens de alto nível e realizar o procedimento de operação em tempo real. Pode ser verificado na tabela da seção 5.6, que o custo computacional para a implementação do algoritmo WM, e utilizando em conjunto o filtro de Kalman (WM+K) e o método de otimização PSO (WM+K+PSO) foi razoavelmente baixo. Comparado aos custos computacionais da realização do algoritmo WM, ao se incorporar o estágio de filtragem de Kalman ocorreu, em média, um acréscimo de 1% na carga computacional.

Em seguida, ao adicionar a otimização de trajetórias por enxame de partículas, o acréscimo médio do custo computacional foi em torno de 15%. Utilizando estes dados como prerrogativa, conclui-se que a possibilidade de implementações em tempo real seja viável. Algumas estratégias para desenvolvimento em tempo real já estão em processo de desenvolvimento e avaliação, permitindo que no futuro a verificação destes algoritmos em situações reais, que envolvam metodologias de controle servovisual.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] S. Hutchinson, G. D. Hager, and P. I. Corke, “A tutorial on visual servo control,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.
- [2] N. Papanikolopoulos and P. Khosla, “Shared and traded telerobotic visual control,” in *IEEE International Conference on Robotics and Automation*, vol. 1, May 1992, pp. 878–885.
- [3] Y. Sumi, Y. Ishiyama, and F. Tomita, “Robot-vision architecture for real-time 6-dof object localization,” *Computer Vision and Image Understanding*, vol. 105, pp. 218–230, 2007.
- [4] H. Shen, G. Nelson, S. Kennedy, D. Nelson, J. Johnson, D. Spiller, M. R. White, and D. B. Kell, “Automatic tracking of biological cells and compartments using particle filters and active contours,” *Chemometrics and Intelligent Laboratory Systems*, vol. 82, no. 1-2, pp. 276–282, May 2006.
- [5] R. Urtasun, D. J. Fleet, and F. P., “Temporal motion models for monocular and multi-view 3d human body tracking,” *Computer Vision and Image Understanding*, vol. 104, pp. 157–177, 2006.
- [6] W. G. Yau, L.-C. Fu, and D. Liu, “Design and implementation of visual servoing system for realistic air target tracking,” in *IEEE International Conference on Robotics and Automation - ICRA*, vol. 1, 2001, pp. 229–234.
- [7] O. Rostamianfar, F. Janabi-Sharifi, and I. Hassanzadeh, “Visual tracking system for dense traffic intersections,” in *Canadian Conference on Electrical and Computer Engineering - CCECE '06.*, May 2006, pp. 2000–2004.
- [8] Z. Gong, W. Ding, and H. Zou, “Data-logging and monitoring of production auto-lines based on visual-tracking tech,” in *IEEE 32nd Annual Conference on Industrial Electronics, IECON 2006*, 2006, pp. 5468–5473.
- [9] K. C. S. Phimoltares, C. Lursinsap, “Face detection and facial feature localization without considering the appearance of image context,” *Image and Vision Computing*, vol. 25, no. 5, pp. 741–753, May 2007.
- [10] R. E. Kalman, “A new approach to linear filtering and prediction problems,” in *Journal of Basic Engineering*, 1960, pp. 35–45.
- [11] A. Gilbert, M. Giles, G. Flachs, R. Rogers, and Y. Hsun, “A real-time video tracking system,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, pp. 47–56, January 1980.

- [12] R. Luo, R. Mullen, and D. Wessell, “An adaptive robotic tracking system using optical flow,” in *IEEE International Conference on Robotics and Automation*, vol. 1, April 1988, pp. 568 – 573.
- [13] N. Papanikolopoulos, P. K. Khosla, and T. Kanade, “Vision and control techniques for robotic visual tracking,” in *International Conference on Robotics and Automation*, Sacramento, CA, Abril 1991.
- [14] F. B. Vidal, “Estudo do movimento em imagens através do fluxo Óptico determinado pelos métodos diferencial, de similaridade de regiões e de filtragem espaço-temporal,” Master’s thesis, Departamento de Engenharia Elétrica - ENE, Faculdade de Tecnologia - FT, Universidade de Brasília - UnB, Março 2005.
- [15] C.-Y. Tsai and K.-T. Song, “Robust visual tracking control of mobile robots based on an error model in image plane,” in *IEEE International Conference Mechatronics and Automation*, vol. 3, 2005, pp. 1218–1223.
- [16] K. T. Song and F. Sheu, “Visual guidance control based on velocity mapping,” in *3rd IFAC Symposium on Intelligent Autonomous Vehicles*, March 1998, pp. 193–198.
- [17] C.-M. Huang, C.-W. Lai, and L.-C. Fu, “Visual tracking with probabilistic data association filter based on the circular hough transform,” in *Proceedings IEEE International Conference on Robotics and Automation, ICRA 2006.*, May 2006, pp. 4094 – 4099.
- [18] R. C. Gonzalez and R. D. Woods, *Processamento de Imagens Digitais*. Edgard Blucher, 2000.
- [19] I.-H. Whang and W.-S. Ra, “Multiple hypothesis tracking in generalized hough transformed images,” in *45th IEEE Conference on Decision and Control*, IEEE, Ed., December 2006, pp. 2134–2139.
- [20] Y. Kosuge, M. Kojima, and S. Tsujimichi, “Multiple manoeuvre model track-oriented mht (multiple hypothesis tracking),” in *Proceedings of the 38th Annual Conference SICE 1999*, July 1999, pp. 1129–1134.
- [21] G. D. Hager and P. N. Belhumeur, “Efficient region tracking with parametric models of geometry and illumination,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ser. 10, vol. 20, 1998, pp. 1025–1039.
- [22] H. Veeraraghavan, P. Schrater, and N. Papanikolopoulos, “Robust target detection and tracking through integration of motion, color and geometry,” *Computer Vision and Image Understanding*, vol. 103, no. 2, pp. 121–138, August 2006.
- [23] R. B. Babu, P. Perez, and P. Bouthemy, “Robust tracking with motion estimation and local kernel-based color modeling,” *Image and Vision Computing*, vol. 25, pp. 1205–1216, 2007.

- [24] P. Brassnet, L. Mihaylova, D. Bull, and N. Canagarajah, "Sequential monte carlo tracking by fusing multiple cues in video sequences," *Image and Vision Computing*, vol. 25, no. 8, pp. 1217–1227, 2007.
- [25] S. S. Tan and D. P. Hart, "A fast and robust feature-based 3d algorithm using compress image correlation," *Pattern Recognition Letters*, vol. 26, no. 11, pp. 1620–1631, August 2005.
- [26] E. Goubaru and M. Sugisaka, "Visual tracking in real-time processing," in *International Joint Conference SICE-ICASE*, October 2006, pp. 5296 – 5299.
- [27] F. B. Vidal and V. H. C. Alcalde, "Motion segmentation in sequential images based on the differential optical flow," *2nd International Conference on Informatics in Control, Automation and Robotics -ICINCO*, pp. 94–100, 2005.
- [28] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," in *International Journal of Computer Vision*, vol. 2, 1989, pp. 283–310.
- [29] D. A. Forsyth and J. Ponce, *Computer Vision: A modern approach*. Prentice-Hall, 2003.
- [30] B. P. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–204, 1981.
- [31] J. Barron, D. J. Fleet, and S. S. Beauchemin, "Systems and experiment performance of optical flow techniques," in *International Journal of Computer Vision*, ser. 1, vol. 12, 1994.
- [32] H. H. Nagel, "Displacement vectors derived from second-order intensity variations in images sequences," in *CVGIP*, vol. 21, no. 1, January 1983, pp. 85–117.
- [33] J. Canny, "A computational approach to edge detection," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
- [34] M. L. A. D. T. D'Orazio, C. Guaragnella, "A new algorithm for ball recognition using circle hough transform and neural classifier," *Pattern Recognition*, vol. 37, no. 3, pp. 393–408, March 2004.
- [35] L. Kitchen and A. Rosenfeld, "Gray level corner detection," *Pattern Recognition Letters*, vol. 1, pp. 95–102, 1982.
- [36] B. Zitova, J. Kautsky, G. Peters, and J. Flusser, "Robust detection of significant points in multiframe images," *PRL*, vol. 20, no. 2, pp. 199–206, February 1999.
- [37] A. Giachetti, "Matching techniques to compute image motion," *Image and Vision Computing*, vol. 18, no. 3, pp. 247–260, February 2000.

- [38] F. B. Vidal and V. H. C. Alcalde, “Window-matching techniques with Kalman filtering for an improved object visual tracking,” in *Third annual IEEE Conference on Automation Science and Engineering, IEEE CASE 2007*, IEEE, Ed., vol. 22-25, September 2007, pp. 829–834.
- [39] T. Dorazio, M. Leo, P. Spagnolo, P. Mazzeo, N. Mosca, and M. Nitti, “A visual tracking algorithm for real time people detection,” in *IEEE - Eight International Workshop on Image Analysis for Multimedia Interactive Services(WIAMIS’07)*, 2007.
- [40] Y. Bar-Shalom and W. D. Blair, *Multitarget-Multisensor Tracking*. Artec House, August 2000, vol. 3.
- [41] H. A. P. Blom and Y. Bar-Shalom, “The interacting multiple model algorithm for systems with markovian switching systems,” *IEEE Transactions on Automatic Control Systems*, vol. 33, no. 8, pp. 780–783, August 1988.
- [42] Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking: Principles, Techniques and Software*. Artech House, 1993.
- [43] J. Collins and J. Uhlmann, “Efficient gating in data association with multivariate gaussian distributed states,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 3, pp. 909–916, July 1992.
- [44] R. Jonker and A. Volgenant, “A shortest augmenting path algorithm for dense and sparse linear assignment problems,” *Computing*, vol. 38, pp. 325–340, 1987.
- [45] Y. Bar-Shalom and T. E. Fortman, “Joint probabilistic data association for multiple targets in clutter,” in *Proceedings Conference on Information Science and Systems*, Princeton, Ed., March 1980.
- [46] A. Srivastava and X. Liu, “Statistical hypothesis pruning for identifying faces from infrared images,” *Image and Vision Computing*, vol. 21, no. 7, pp. 651–661, July 2003.
- [47] F. T. Ramos, B. Uprocft, S. Kumar, and H. F. Durrant-Whyte, “Recognising and segmenting objects in natural environments,” in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, October 2006, pp. 5866–5871.
- [48] S. Ertürk, “A new perspective to block motion estimation for video compression: High-frequency component matching,” *IEEE Signal Processing Letters*, vol. 14, no. 2, pp. 113–116, February 2007.
- [49] T. Kanade and M. Okutomi, “A stereo matching algorithm with an adaptive window: Theory and experiment,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 9, pp. 920–932, September 1994.

- [50] Y. Han and H. Hahn, "Visual tracking of a moving target using active contour based ssd algorithm," *Robotics and Autonomous Systems*, vol. 53, pp. 265–281, 2005.
- [51] M. Irfan, A. Khan, and H. Jamal, "Fpga based implementation of mpeg-2 compression algorithm," in *The 17th International Conference on Microelectronics, 2005. ICM 2005.*, December 2005, pp. 240–244.
- [52] L. Yufei, F. X., and W. Q., "A high-performance low cost sad architecture for video coding," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 535–541, May 2007.
- [53] J. Vanne, E. Aho, T. Hamalainen, and K. Kuusilinna, "A high-performance sum of absolute difference implementation for motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 7, pp. 876–883, July 2006.
- [54] B. Chaouke, H. Tairi, L. Masmoudi, and L. Radouane, "Pi controller for visual tracking: further results," in *Riding the Crest into the 21st Century OCEANS '99 MTS/IEEE*, vol. 3, September 1999, pp. 1480–1485.
- [55] Y. J. Lee, D.-Y. Kim, and M. J. Chung, "Feature matching in omnidirectional images with a large sensor motion for map generation of a mobile robot," *Pattern Recognition Letters*, vol. 25, pp. 413–427, 2004.
- [56] L. Huiping, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital video," *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 147–156, January 2000.
- [57] G. D. Hager and P. N. Belhumeur, "Real-time tracking of image regions with changes in geometry and illumination," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*. Computer Society Conference on Proceedings CVPR'96, 1996, pp. 403–410.
- [58] J. Banks and M. Bennamoun, "Reliability analysis of the rank transform for stereo matching," in *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, ser. 6, vol. 31, 2001, pp. 870–880.
- [59] K. Yamada, T. Ichikawa, T. Naemura, K. Aizawa, and T. Saito, "Generation of a disparity panorama using a 3-camera capturing system," in *International Conference on Image Processing*, vol. 2, 2000, pp. 772–775.
- [60] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *3rd European Conference Computer Vision*, 1994, pp. 151–158.
- [61] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using hausdorff distance," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ser. 9, vol. 15, 1993, pp. 850–863.

- [62] F. Essannouni, R. Oulad, H. Thami, D. Aboutajdine, and A. Salam, "Optimal frequency ssd matching using cosine kernel," in *International Conference on Electronics, Circuits and Systems, ICECS 2007*, IEEE, Ed., December 2007, pp. 119–122.
- [63] A. Fitch, A. Kadyrov, W. Christmas, and J. Kittler, "Fast robust correlation," *IEEE Transactions on Image Processing*, vol. 14, no. 8, pp. 1063–1073, August 2005.
- [64] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics, Control, Sensing, Vision and Intelligence*. Singapore: McGraw-Hill, Inc., 1987.
- [65] T. Kang-hua, W. Mei-ping, and H. Xiao-ping, "Multiple model Kalman filtering for mems-imu/gps integrated navigation," in *2nd IEEE Conference on Industrial Electronics and Applications - ICIEA 2007.*, Harbin, May 2007, pp. 2062 – 2066.
- [66] C. Antoniou, M. Ben-Akiva, and H. Koutsopoulos, "Nonlinear Kalman filtering algorithms for on-line calibration of dynamic traffic assignment models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 4, pp. 661–670, 2007.
- [67] W. X., Y. qing G., and X. dong Z., "A bank of kalman filters and a robust Kalman filter applied in fault diagnosis of aircraft engine sensor/actuator," in *Second International Conference on Innovative Computing, Information and Control, 2007. ICICIC '07.*, September 2007.
- [68] C.-Y. T., K.-T. S., X. Dutoit, H. Van Brussel, and M. Nuttin, "Robust mobile robot visual tracking control system using self-tuning Kalman Filter," in *International Symposium on Computational Intelligence in Robotics and Automation, 2007 - CIRA 2007*, June 2007, pp. 161–166.
- [69] M. St-Pierre and D. Gingras, "Comparison between the unscented Kalman filter and the extended kalman filter for the position estimation module of an integrated navigation information system," in *IEEE Intelligent Vehicles Symposium*. IEEE, June 2004, pp. 831–835.
- [70] B. Anderson and J. Moore, *Optimal Filtering*, T. Kailath, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1979.
- [71] G. Welch and G. Bishop, "An introduction to the Kalman filter," Department of Computer Science, University of North Carolina at Chapell Hill, Tech. Rep., 2007.
- [72] A. Gelb, *Applied Optimal Estimation*. MIT Press, 1974.
- [73] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice*. Upper Saddle River, NJ: Prentice-Hall, 1993.
- [74] P. S. Maybeck, *Stochastic Models, Estimation and Control, Volume 1*. Academic Press, Inc., 1979.

- [75] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 2nd ed. John Wiley & Sons, Inc., 1992.
- [76] S. Park and A. J. K., “Simultaneous tracking of multiple body parts of interacting persons,” *Computer and Vision and Image Understanding*, vol. 102, pp. 1–21, 2006.
- [77] E. Loutas, C. Nikou, and I. Pitas, “Information theory-based analysis of partial and total occlusion in object tracking,” in *International Conference on Image Processing. 2002. Proceedings*, vol. 2, 2002, pp. 309–312.
- [78] N. Kanhere, S. Pundlik, and S. Birchfield, “Vehicle segmentation and tracking from a low-angle off-axis camera,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, vol. 2, June 2005, pp. 1152–1157.
- [79] O. Lanz, “Approximate bayesian multibody tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1436–1449, September 2006.
- [80] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” in *IEEE International Conference on Neural Networks*, Perth, Australia, 1995, pp. 1942–1948.
- [81] M. Clerc and J. Kennedy, “The particle swarm: Explosion, stability and convergence in a multi-dimensional complex space,” *IEEE Transactions Evolution Computational*, vol. 6, no. 1, pp. pp.:58–73, 2002.
- [82] R. Akbari, M. Jazi, and M. Palhang, “A hybrid method for robust multiple objects tracking in cluttered background,” in *Information and Communication Technologies - 2nd ICTTA '06*, vol. 1, 2006, pp. 1562–1567.
- [83] Y. Owechko, S. Medasani, and N. Srinivasa, “Classifier swarms for human detection in infrared imagery,” in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04*, 2004, pp. 121–121.
- [84] L. Anton-Canalis, M. Hernandez-Tejera, and E. Sanchez-Nielsen, “Particle swarms as video sequence inhabitants for object tracking in computer vision,” in *Sixth International Conference on Intelligent Systems Design and Applications, ISDA '06*, vol. 2, 2006, pp. 604–609.
- [85] L. Anton-Canalis, E. Sanchez-Nielsen, and M. Hernandez-Tejera, “A particle swarm approach to visual tracking,” in *VISAPP' 06*, vol. 2, Setúbal, Portugal, February 2006, pp. 221–228.
- [86] G. Venter and J. Sobieszczanski-Sobieski, “Particle swarm optimization,” in *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, vol. AIAA-2002-1235, Denver, CO, U.S., April, 22-25 2002.

- [87] J. Chen, F. Pan, T. Cai, and X. Tu, “Stability analysis of particle swarm optimization without lipschitz constraint,” *Journal of Control Theory and Applications*, vol. 1, no. 1, pp. 86–90, November 2003.
- [88] D.-J. Jwo, “Remarks on the Kalman filtering simulation and verification,” in *Applied Mathematics and Computation*, ser. 1, vol. 186, March 2007, pp. 159–174.
- [89] K. Atteson, “The asymptotic redundancy of bayes rules for markov chains,” *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 2104–2109, 1999.
- [90] S. J. Julier, J. K. Uhlmann, and H. Durrant-White, “A new approach for filtering nonlinear system,” in *Proceedings of American Control Conference*, 1995, pp. 1628–1632.
- [91] S. J. Julier and J. K. Uhlmann, “A general method for approximating nonlinear transformation of probability distributions,” RRG, Dept. of Engineering Science, University of Oxford, Tech. Rep., November 1996.
- [92] J. Joseph and L. V. Jr., “A comparison of unscented and extended Kalman filtering for estimating quaternion motion,” in *Proceedings of American Control Conference*, 2003, pp. 2435–2440.
- [93] Y. Hao, Z. Xiong, F. Sun, and X. Wang, “Comparison of unscented Kalman filters,” in *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation*. Harbin, China: IEEE, August 2007, pp. 895–899.
- [94] G. Casella and C. P. Robert, “Rao-blackwellisation of sampling schemes,” *Biometrika*, vol. 83, pp. 81–94, 1996.
- [95] A. Doucet, N. Freitas, K. Murphy, and S. Russel, “Rao-blackwellised particle filtering for dynamic bayesian networks,” in *Proceedings of the 16th Conference on Uncertainty in Artificial intelligence*, Stanford, 2000, pp. 176–183.
- [96] X. Xinyu and L. Baoxin, “Rao-blackwellised particle filter with adaptive system noise and its evaluation for tracking in surveillance,” in *Proceedings SPIE*, vol. 6077, 2006.

APÊNDICES

A. DIAGRAMA DE BLOCOS DOS ALGORITMOS DESENVOLVIDOS

Durante toda a realização deste trabalho, diversos algoritmos auxiliares foram desenvolvidos. Estes algoritmos fazem parte dos principais códigos de rastreamento desenvolvidos. Entretanto, para melhor esclarecer suas funcionalidades e propriedades, todos foram colocados na forma de diagrama de blocos sequenciais.

A.1 ALGORITMO PARA CÁLCULO DO CENTRÓIDE DE IMAGENS SEGMENTADAS

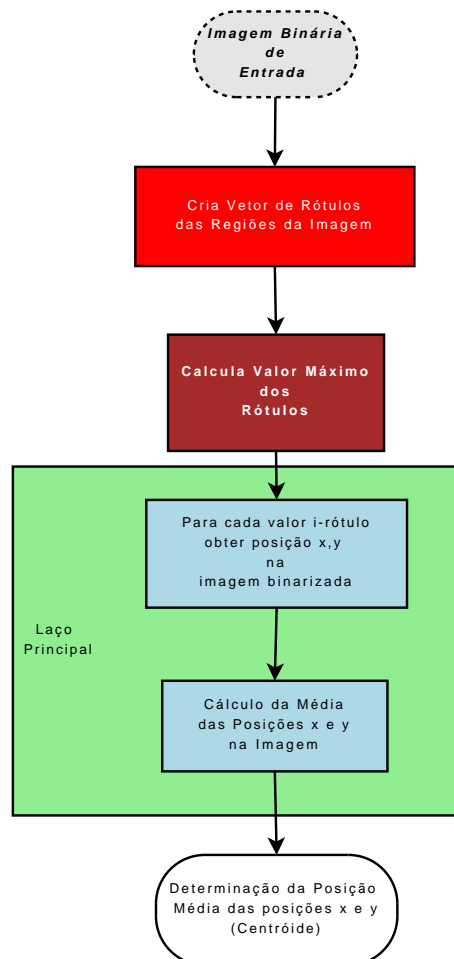


Figura A.1: Diagrama de Blocos do Cálculo do Centróide de Imagens Segmentadas.

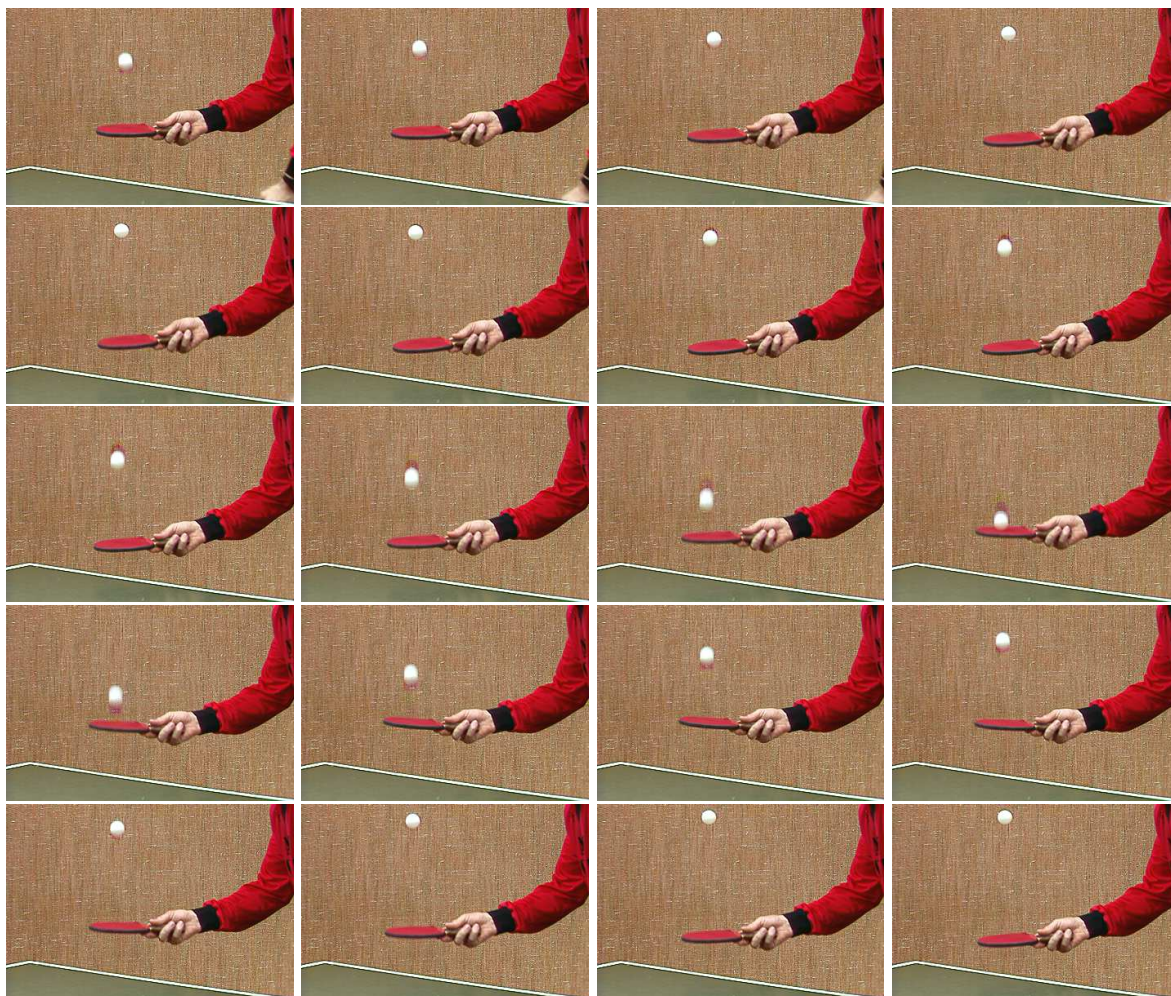
B. SEQÜÊNCIAS DE IMAGENS

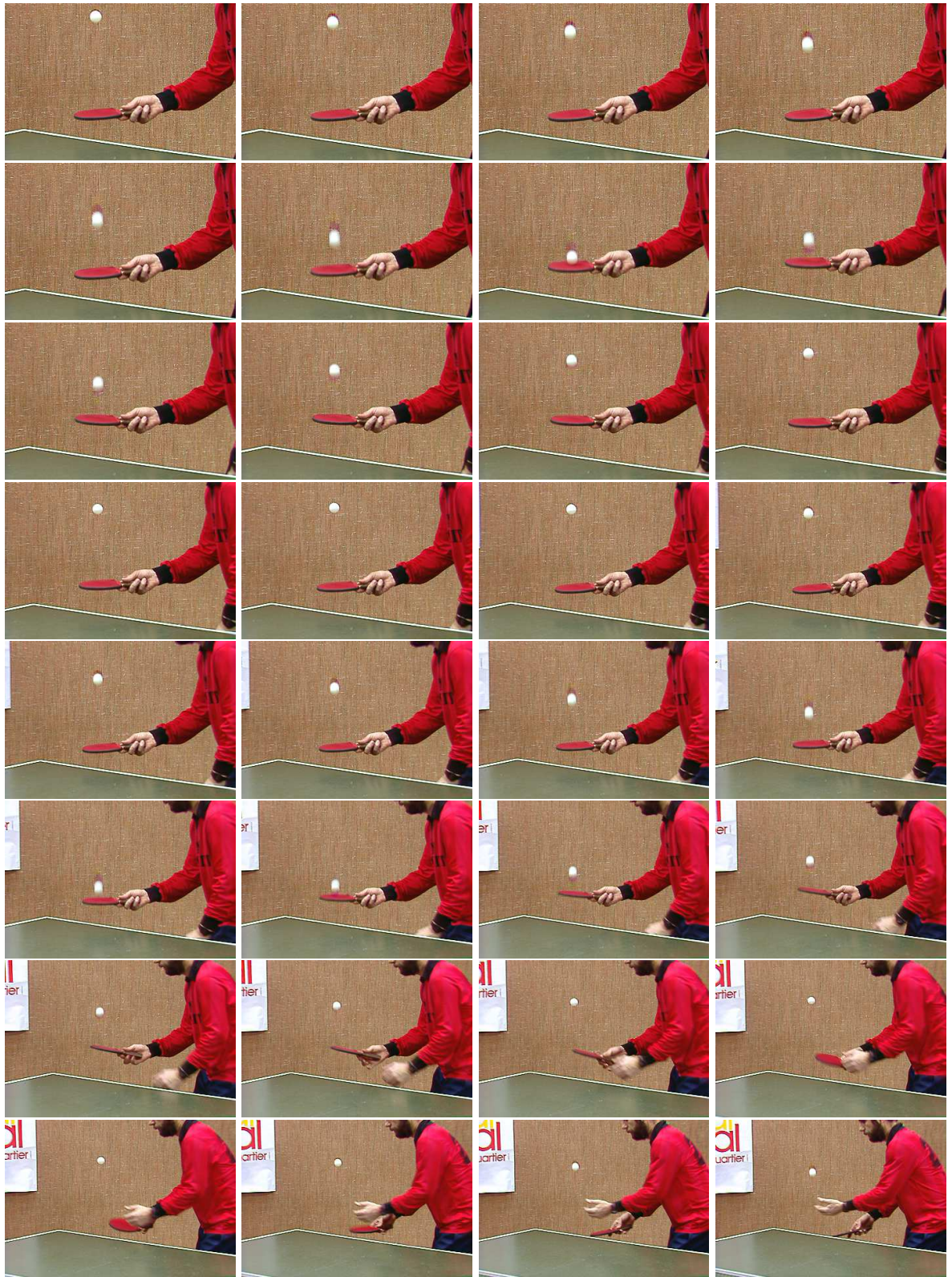
B.1 SEQÜÊNCIA TÊNIS DE MESA

Informações:

Dimensões espaciais (H × W): 352 × 240

Formato: PNG



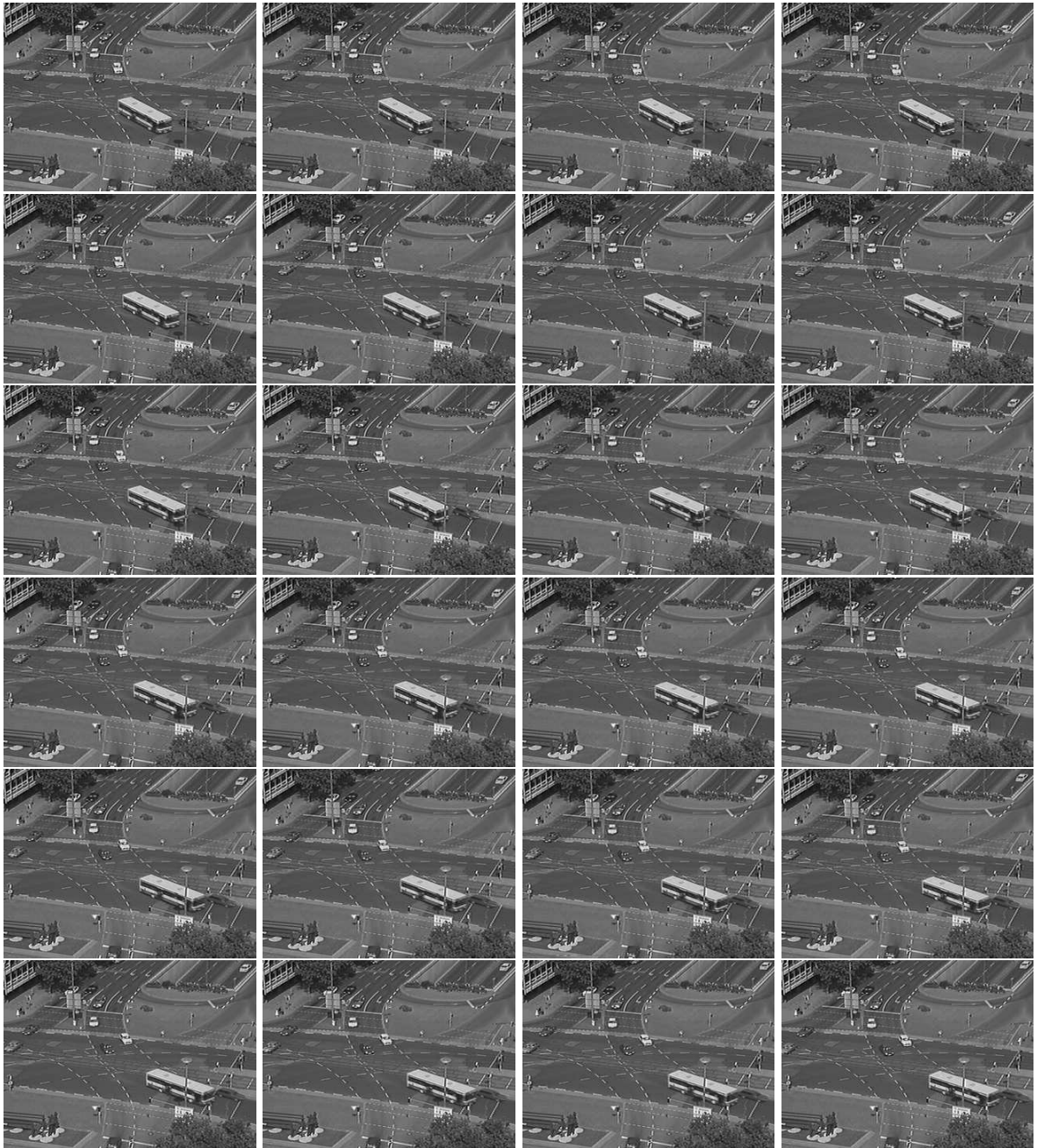


B.2 SEQUÊNCIA TRÂNSITO URBANO

Informações:

Dimensões espaciais ($H \times W$): 240 \times 320

Formato: JPEG



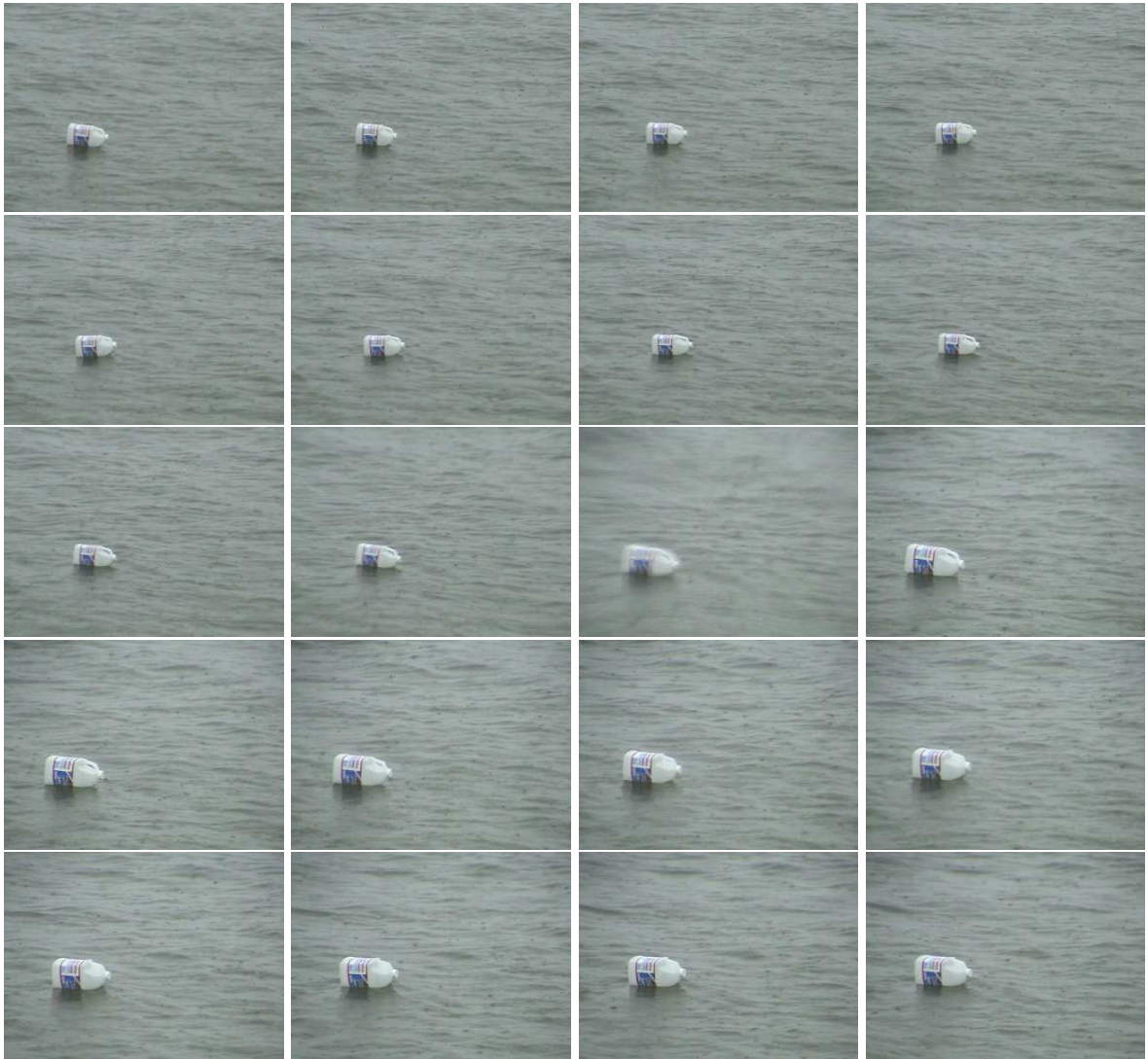


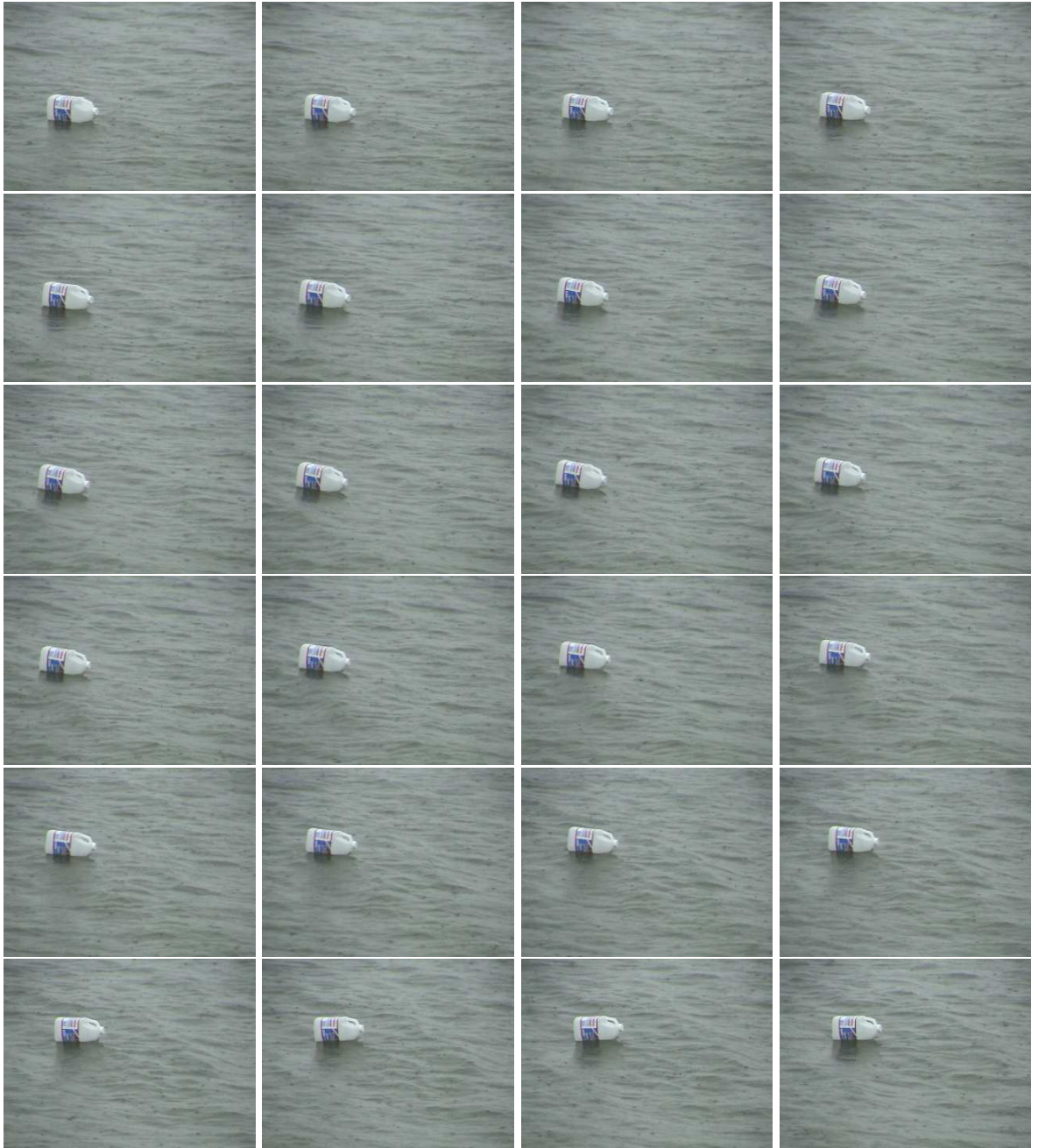
B.3 SEQUÊNCIA GARRAFA FLUTUANDO NO MAR

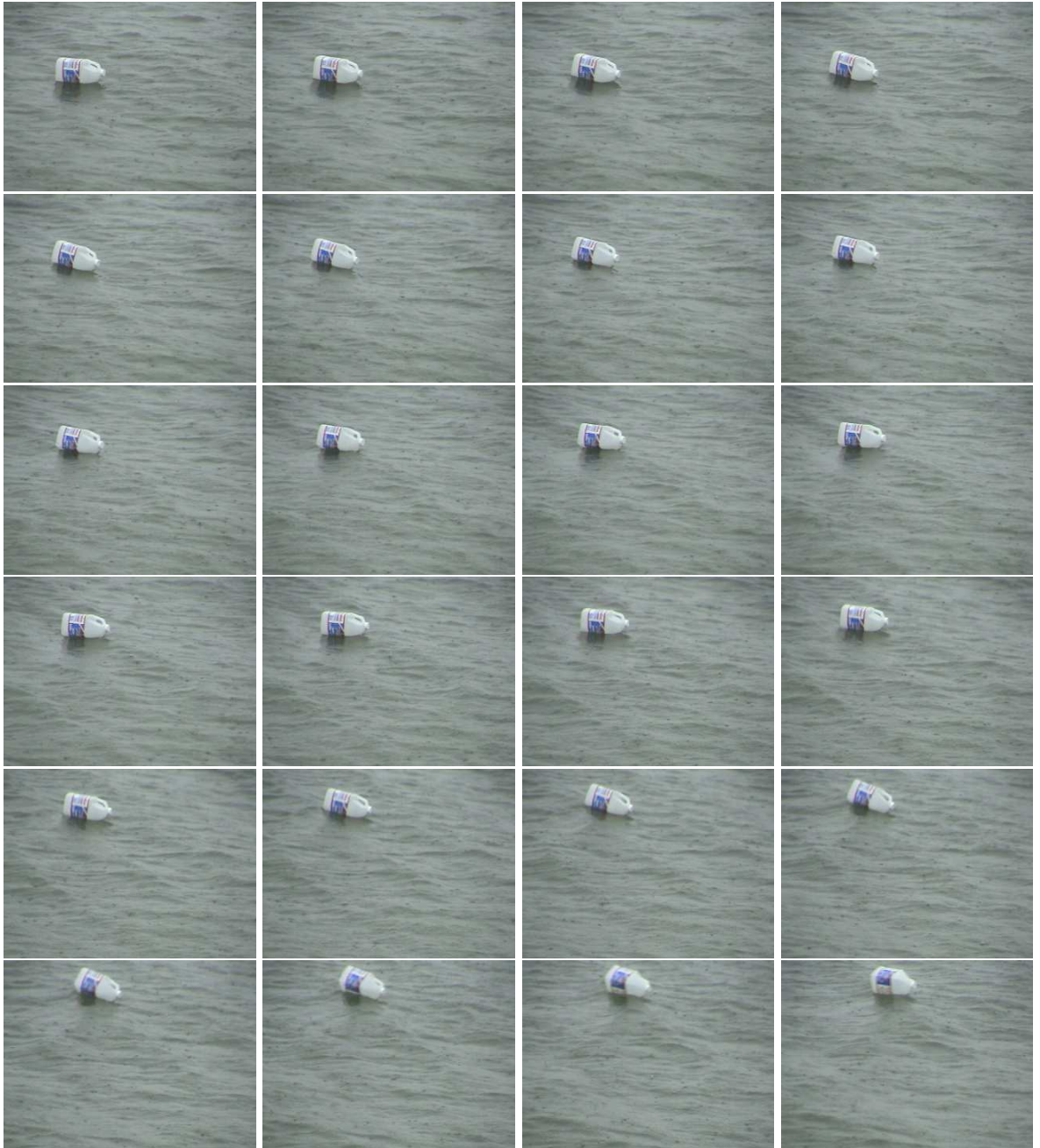
Informações:

Dimensões espaciais (H × W): 320 × 240

Formato: BMP







B.4 SEQUÊNCIA ENCONTRO NO SHOPPING

Informações:

Dimensões espaciais (H × W): 384 × 288

Formato: JPG







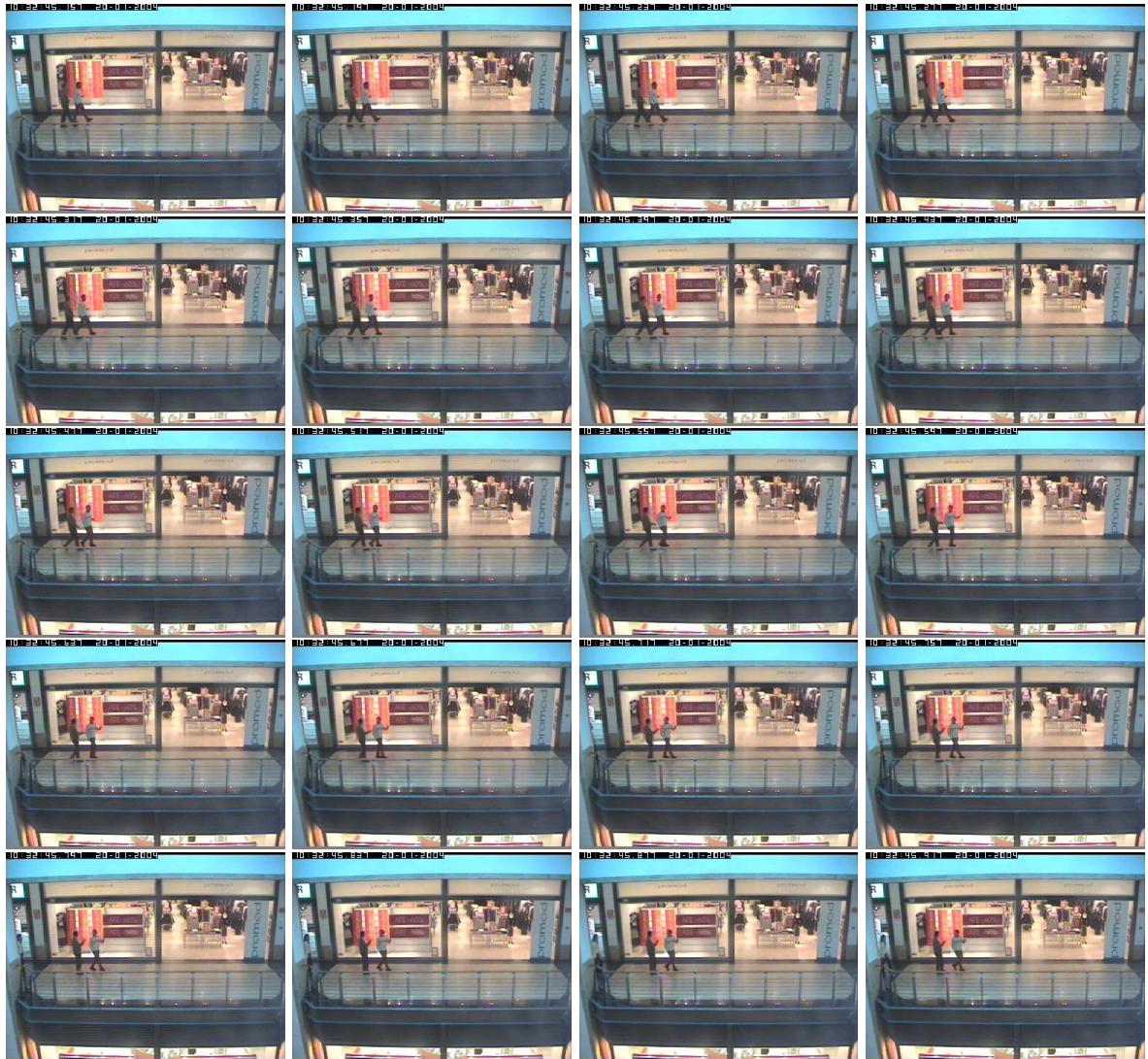


B.5 SEQUÊNCIA ENCONTRO NO SHOPPING COM OCLUSÃO PARCIAL

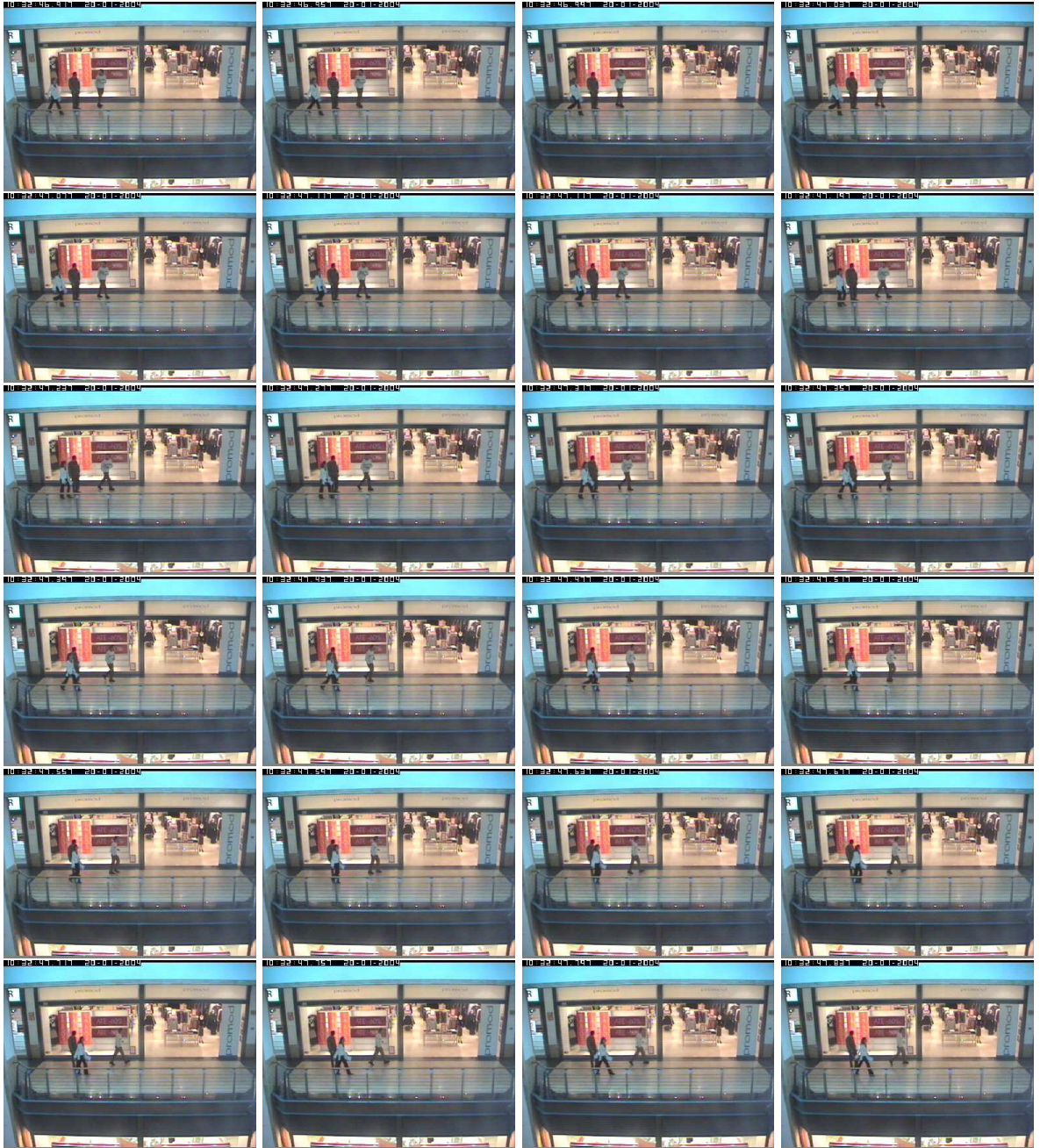
Informações:

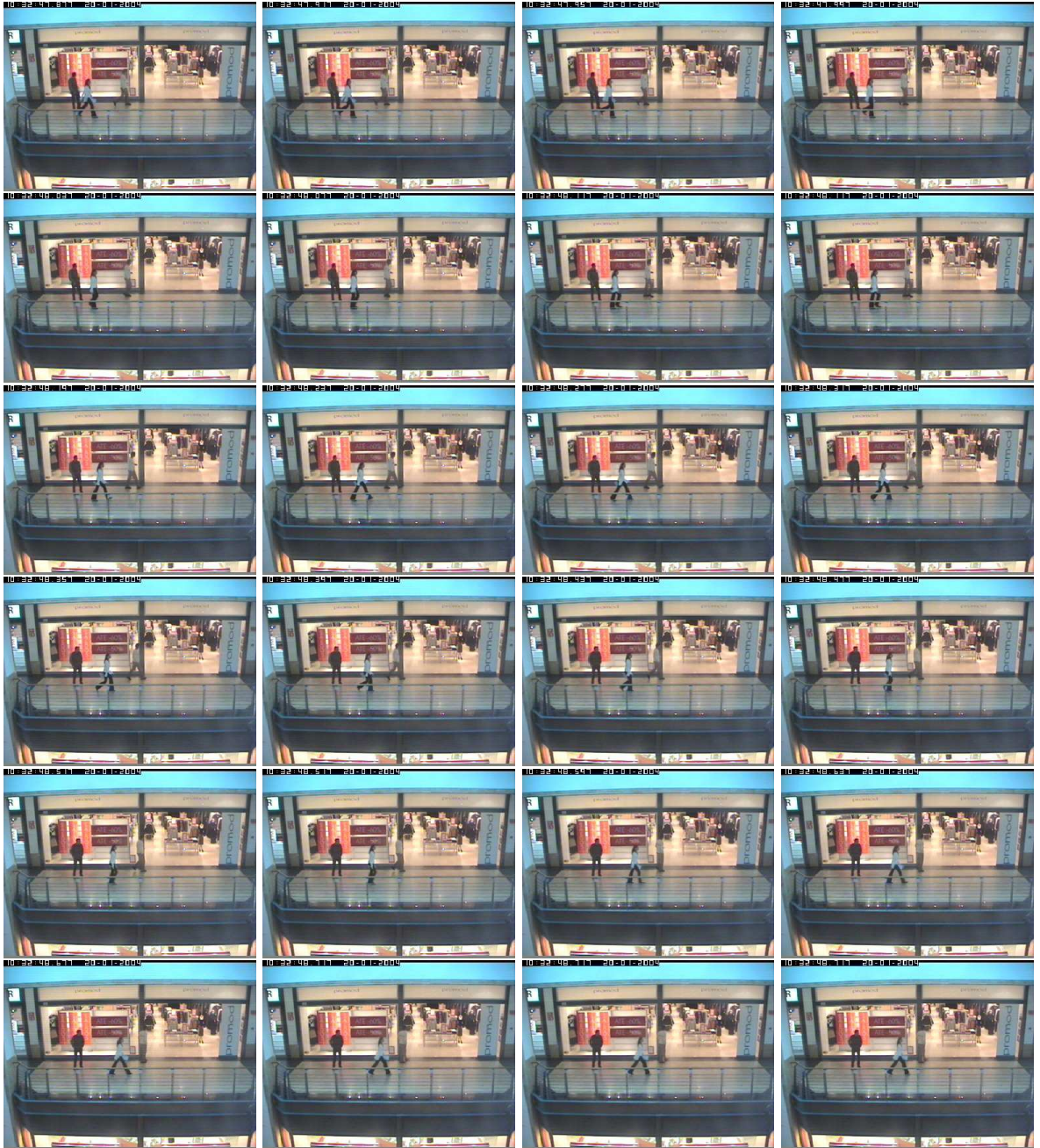
Dimensões espaciais ($H \times W$): 384 \times 288

Formato: JPG







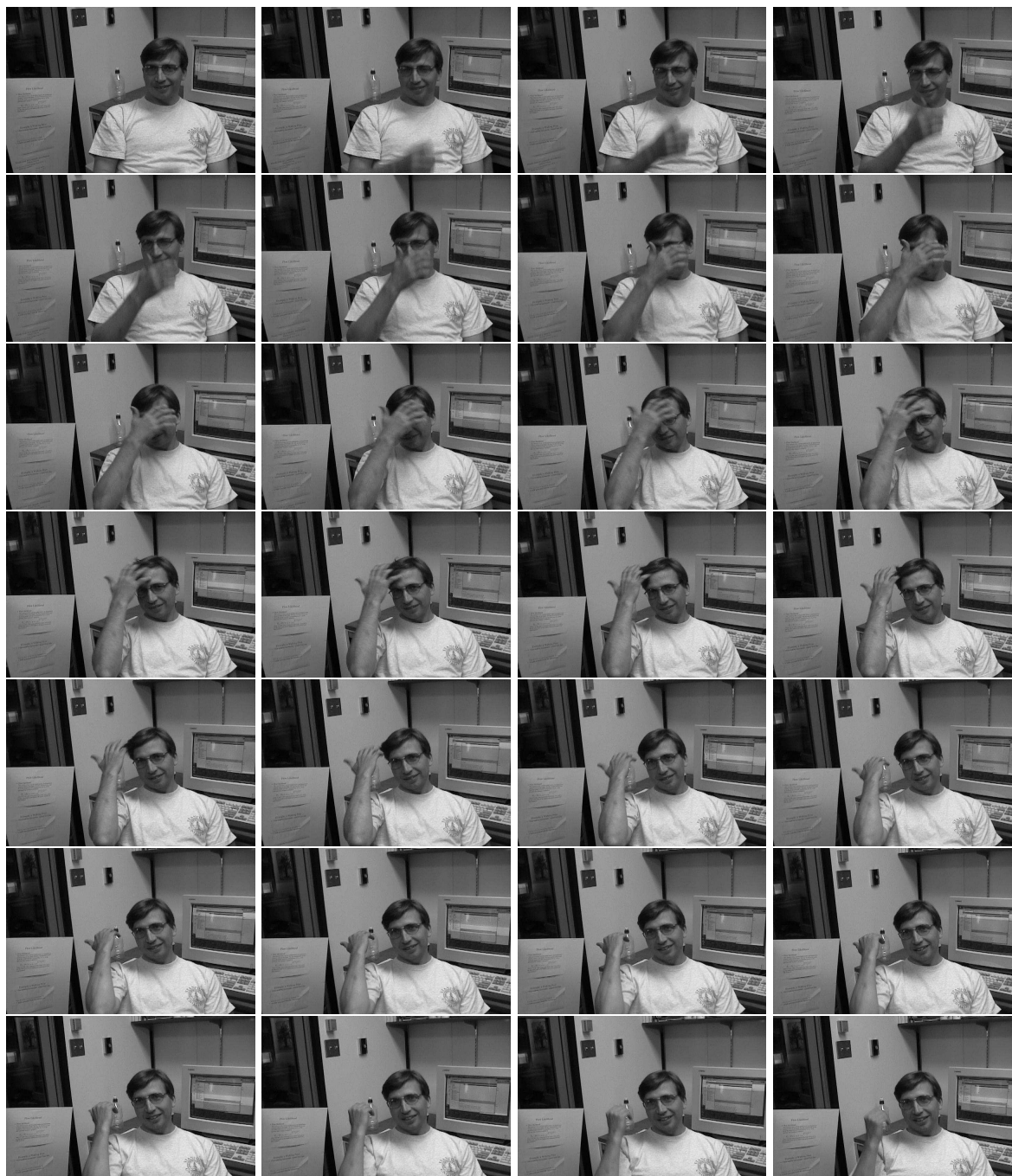


B.6 SEQUÊNCIA DE FACE HUMANA COM OCLUSÕES PARCIAIS

Informações:

Dimensões espaciais ($H \times W$): 480 × 720

Formato: JPG





B.7 SEQUÊNCIA OBJETO COM MOVIMENTOS DIVERSOS

Informações:

Dimensões espaciais (H × W): 240 × 320

Formato: JPG





C. VARIACÕES DO FILTRO DE KALMAN

Nesta seção é apresentada uma breve descrição das principais variações do Filtro de Kalman, seguindo a abordagem proposta pelo artigo fundamental de R. E. Kalman em 1960 [10]. As principais formas variantes são: Filtro de Kalman Estendido (*Extended Kalman Filter - EKF*) e Filtro de Kalman Unscented (*Unscented Kalman Filter - UKF*).

C.1 FILTRO DE KALMAN ESTENDIDO

Seguindo a formulação apresentada na seção 4.2.1, em que o Filtro de Kalman tem como objetivo a estimação do vetor de estados $\mathbf{x} \in \mathbb{R}^n$ de um processo de tempo discreto e descrito por uma *equação a diferenças finitas estocástica linear*, contradiz em todos os aspectos as aplicações em processos reais, onde quase sua totalidade não-linear, obrigando a linearização destes, para terem sua aplicabilidade alcançada. Sob esta influência surge o conceito do Filtro de Kalman em sua forma estendida, em que lineariza-se a média e covariância atual, dispensando linearizações tradicionais (ex.: Séries de Taylor). A seguir são apresentadas as principais equações e formulações do Filtro de Kalman Estendido.

C.1.1 Equações do Filtro de Kalman Estendido

Assume-se que o processo em análise tem um vetor de estado $\mathbf{x} \in \mathbb{R}^n$, conforme já foi citado anteriormente, mas neste caso é descrito pela equação *a diferenças finitas estocástica não-linear*.

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}), \quad (\text{C.1})$$

com vetor de medição $\mathbf{z} \in \mathbb{R}^m$ dado por

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k), \quad (\text{C.2})$$

em que as matrizes de variáveis aleatórias \mathbf{w}_k e \mathbf{v}_k novamente representam os ruídos de processo e de medições respectivamente. Neste caso a função não-linear f (Eq. C.1) é descrita em função do instante de tempo anterior ($k - 1$) para se obter o estado no tempo atual k , incluindo o vetor de entrada do processo (\mathbf{u}_{k-1}) e o ruído de processo com média-zero \mathbf{w}_k . A função não-linear de medição h correlaciona o estado \mathbf{x}_k com a medição \mathbf{z}_k .

Na prática, pode-se aproximar o vetor de estados e de medições sem as variáveis de ruído de processo e medição, por serem modeladas por uma distribuição normal gaussiana. Têm-se

então que as variáveis de estado e medição serão:

$$\tilde{\mathbf{x}}_k = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, 0) \quad (\text{C.3})$$

e

$$\tilde{\mathbf{z}}_k = h(\tilde{\mathbf{x}}_k, 0) \quad (\text{C.4})$$

onde $\hat{\mathbf{x}}$ é a estimação *a posteriori* do estado. De uma forma simplificada pode-se afirmar que o Filtro de Kalman Estendido é um estimador de estados "ad hoc" que faz uma aproximação otimizada da *Regra de Bayes* por linearização [89].

A realização computacional do filtro, pode ser descrita utilizando como base as equações que linearizam as estimações das equações C.3 e C.4.

$$\mathbf{x}_k \approx \tilde{\mathbf{x}}_k + \mathbf{A}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \mathbf{W}\mathbf{w}_{k-1}, \quad (\text{C.5})$$

$$\mathbf{z}_k \approx \tilde{\mathbf{z}}_k + \mathbf{H}(\mathbf{x}_k - \tilde{\mathbf{x}}_k) + \mathbf{V}\mathbf{v}_k \quad (\text{C.6})$$

onde

- \mathbf{x}_k e \mathbf{z}_k são os vetores de estado e medições atuais respectivamente;
- $\tilde{\mathbf{x}}_k$ e $\tilde{\mathbf{z}}_k$ são os vetores de estado e medições aproximados (vide Eqs. C.3 e C.4);
- $\hat{\mathbf{x}}_k$ é uma estimação *a posteriori* do estado no instante k ;
- as variáveis aleatórias \mathbf{w}_k e \mathbf{v}_k representam os ruídos de processo e de medição;
- \mathbf{A} é a matriz *Jacobiana* das derivadas parciais da função f em relação à vetor de estados \mathbf{x} , dado como:

$$\mathbf{A}_{[i,j]} = \frac{\partial f_{[i]}}{\partial \mathbf{x}_{[j]}}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, 0); \quad (\text{C.7})$$

- \mathbf{W} é a matriz *Jacobiana* das derivadas parciais da função f em relação a w :

$$\mathbf{W}_{[i,j]} = \frac{\partial f_{[i]}}{\partial \mathbf{w}_{[j]}}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, 0); \quad (\text{C.8})$$

- \mathbf{H} é a matriz *Jacobiana* de derivadas parciais de h com relação a \mathbf{x} :

$$\mathbf{H}_{[i,j]} = \frac{\partial h_{[i]}}{\partial \mathbf{x}_{[j]}}(\tilde{\mathbf{x}}_k, 0); \quad (\text{C.9})$$

- \mathbf{V} é a matriz *Jacobiana* de derivadas parciais de h em relação a \mathbf{v}

$$\mathbf{V}_{[i,j]} = \frac{\partial h_{[i]}}{\partial \mathbf{v}_{[j]}}(\tilde{\mathbf{x}}_k, 0); \quad (\text{C.10})$$

Com a finalidade de facilitar a notação, não foi utilizado o subscrito k nos Jacobianos \mathbf{A} , \mathbf{W} , \mathbf{H} e \mathbf{V} . Deve-se também observar que os valores destes Jacobianos são modificados a cada passo k de interação do algoritmo de realização do filtro.

Realizando a mesma abordagem relativa ao erro de predição, de forma análoga pode-se obter as equações de realização do Filtro de Kalman Estendido, conforme apresentado nas equações a seguir.

- Equações de Atualização do Tempo

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, 0) \quad (\text{C.11})$$

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{W}_k \mathbf{Q}_{k-1} \mathbf{A}_k^T \quad (\text{C.12})$$

- Equações de Atualização das Medições

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T)^{-1} \quad (\text{C.13})$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-, 0)) \quad (\text{C.14})$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (\text{C.15})$$

Análogo ao Filtro de Kalman em sua forma linear, as equações C.13, C.14 e C.15 realizam a atualização das medições corrigindo o estado e as estimações das covariâncias com a medição \mathbf{z}_k . Deve-se verificar que a realização do algoritmo da forma estendida do Filtro de Kalman é a mesma da forma linear, entretanto devido que a cada instante de tempo k novas matrizes de Jacobianos ($\mathbf{A}_{[i,j]}$, $\mathbf{W}_{[i,j]}$, $\mathbf{H}_{[i,j]}$ e $\mathbf{V}_{[i,j]}$) deverão ser calculadas, a carga computacional necessária para realização da equação do filtro se torna elevada, prejudicando a aplicação em situações de controle em tempo real.

C.2 FILTRO DE KALMAN UNSCENTED

O *Filtro de Kalman Unscented* - UKF - é um filtro não-linear, inicialmente proposto por Julier e Uhlman ([90] e [91]), utilizando modelos totalmente não-lineares e aproximações por distribuições Gaussianas do estado da variável aleatória. Diferentemente da forma Estendida do Filtro de Kalman, em que o princípio se faz por aproximação do modelo não-linear por séries de Taylor (ver Seção C.1), a forma *Unscented* ainda necessita um conjunto mínimo de pontos de amostragens (i. e. *sigma points*) que são escolhidos de forma cuidadosa para permitir que a média e a covariância *a posteriori* sejam obtidas de forma precisa para aproximações de segunda ordem para qualquer não-linearidade, evitando o cálculo do Jacobiano. Partindo do pressuposto que a variável aleatória é modelada por uma distribuição Gaussiana, permite que a média e a covariância *a posteriori* possam ser calculadas com precisão até a terceira ordem. As equações do Filtro de Kalman *Unscented* são apresentadas a seguir.

Para obter a precisão e acuidade da matriz de covariância dos vetores de estado e medições, faz-se necessário para o Filtro de Kalman Unscented estender as dimensões para o vetor de estados e sua matriz de covariância. A inicialização do filtro será dada por:

$$\bar{\mathbf{x}}_0^a = E(\mathbf{x}_0) = \begin{bmatrix} \bar{\mathbf{x}}_0^T & 0 & 0 \end{bmatrix}^T \quad (\text{C.16})$$

$$\bar{\mathbf{P}}_0^a = E((\mathbf{x}_0 - \bar{\mathbf{x}}_0)(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^T) = \begin{bmatrix} \bar{\mathbf{P}}_0 & 0 & 0 \\ 0 & \mathbf{Q}_0 & 0 \\ 0 & 0 & \mathbf{R}_0 \end{bmatrix} \quad (\text{C.17})$$

em que $\bar{\mathbf{x}}_0^a$ e $\bar{\mathbf{P}}_0^a$ denotam a média e a covariância do vetor de estados \mathbf{x}_0 e covariância de \mathbf{P}_0 no instante inicial de realização do algoritmo. O vetor que apresenta os *sigma points*, bem como as etapas de atualização do tempo e das medições são descritas a seguir.

- *Cálculo dos Sigma Points*

$$\chi_{k-1}^a = \begin{bmatrix} \bar{\mathbf{x}}_{k-1}^a & \bar{\mathbf{x}}_{k-1}^a + \sqrt{(L + \lambda)\bar{\mathbf{P}}_{k-1}^a} & \bar{\mathbf{x}}_{k-1}^a - \sqrt{(L + \lambda)\bar{\mathbf{P}}_{k-1}^a} \end{bmatrix} \quad (\text{C.18})$$

L é a dimensão de $\bar{\mathbf{x}}_0^a$ e λ um fator de escala.

- *Atualização do Tempo*

$$\chi_k^x = f(\chi_{k-1}^x, \chi_{k-1}^v) \quad (\text{C.19})$$

em que o processo utilizado é não-linear conforme descrito pelas equações C.1 e C.2.

$$\hat{\mathbf{x}}_k = \sum_{i=0}^{2L} \mathbf{W}_i^{(m)} \chi_{i,k}^x \quad (\text{C.20})$$

$\mathbf{W}^{(m)}$ é a matriz de pesos ponderados em torno do vetor \bar{x} com valores distribuídos simetricamente.

$$\hat{\mathbf{P}}_k = \sum_{i=0}^{2L} \mathbf{W}_i^{(c)} (\chi_k^x - \hat{\mathbf{x}}_k) (\chi_k^x - \hat{\mathbf{x}}_k)^T \quad (\text{C.21})$$

$$\mathbf{Z}_k = h(\chi_k^x, \chi_{k-1}^v) \quad (\text{C.22})$$

$$\hat{\mathbf{z}}_k = \sum_{i=0}^{2L} \mathbf{W}_i^{(m)} \mathbf{Z}_{i,k} \quad (\text{C.23})$$

- *Atualização das Medições*

$$\hat{\mathbf{P}}_{zz,k} = \sum_{i=0}^{2L} \mathbf{W}_i^{(c)} (\mathbf{Z}_{i,k} - \hat{\mathbf{z}}_k) (\mathbf{Z}_{i,k} - \hat{\mathbf{z}}_k)^T \quad (\text{C.24})$$

$$\hat{\mathbf{P}}_{xz,k} = \sum_{i=0}^{2L} \mathbf{W}_i^{(c)} (\chi_{i,k}^x - \hat{\mathbf{x}}_k) (\mathbf{Z}_{i,k} - \hat{\mathbf{z}}_k)^T \quad (\text{C.25})$$

$$\mathbf{K}_k = \hat{\mathbf{P}}_{xz,k} \hat{\mathbf{P}}_{zz,k}^{-1} \quad (\text{C.26})$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k) \quad (\text{C.27})$$

$$\hat{\mathbf{P}}_k = \hat{\mathbf{P}}_{k-1} - \mathbf{K}_k \hat{\mathbf{P}}_{zz,k} \mathbf{K}_k^T \quad (\text{C.28})$$

Em que α , β e κ são parâmetros de escala definidos por:

$$\begin{aligned} \lambda &= \alpha^2(n + \kappa) - n \\ \mathbf{W}_0^{(m)} &= \frac{\lambda}{(L+\lambda)} \\ \mathbf{W}_0^{(c)} &= \frac{\lambda}{(L+\lambda)} + (1 - \alpha^2 + \beta) \\ \mathbf{W}_i^{(m)} = \mathbf{W}_i^{(c)} &= \frac{1}{2(L+\lambda)}, \quad i = 1, 2, \dots, 2L. \end{aligned}$$

Pode-se verificar, que o conjunto de equações apresentadas possuem a mesma estrutura do Filtro de Kalman em sua forma linear. A diferença se faz na utilização da *Transformação Unscented* da média e da covariância nas etapas de atualização, bem como a aplicação de métodos de funções estatísticas ponderadas para computar as matrizes de covariância $\hat{\mathbf{P}}_k$, $\hat{\mathbf{P}}_{zz,k}$ e $\hat{\mathbf{P}}_{xz,k}$.

A vantagem da forma *Unscented*, se comparado com a forma Estendida, é a baixa carga computacional e a fácil implementação em aplicações voltadas para a estimação de estados, estimação de parâmetros e treinamento de redes neurais [92]. Segundo o trabalho de

Y. Hao [93], pode-se ainda obter diversas forma variantes da versão *Unscented* do Filtro de Kalman, como: *Filtro de Kalman Unscented Aditivo (AUKF)*, *Filtro de Kalman Unscented Rao-Blackwellised (RBUKF)* e *Filtro de Kalman Unscented Rao-Blackwellised Aditivo (RBAUKF)*. A forma Aditiva do Filtro de Kalman Unscented consiste em uma versão generalizada da transformação *Unscented*, com a diferença que agora o ruído de processo (\mathbf{w}_k) e de medição (\mathbf{v}_k) são fatores que são adicionados ao modelo não-linear, e não fazendo parte diretamente da função que modelo o processo (Eqs. C.1 e C.2), como é apresentado nas equações C.29 e C.30 abaixo.

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \quad (\text{C.29})$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (\text{C.30})$$

Em que as funções f e h possuem as mesmas características não-lineares, conforme foram anteriormente descritas na Seção C.1.1.

Na forma *Rao-Blackwellised do Filtro de Kalman Unscented*, são marginalizadas¹ variáveis através da aplicação da metodologia Rao-Blackwell ([94] e [95]). Esta metodologia é uma forma efetiva de redução no tamanho (ex: número de variáveis de estado) em um filtro de Partículas ([96]), que também podem ser utilizados na redução das amostras iniciais (*sigma points*) do Filtro de Kalman Unscented.

¹Variáveis marginalizadas são aquelas que passam a expressar menor importância no processo, sendo excluídas e/ou desprezadas.