



**PROPOSTA DE UM FILTRO DE PARTÍCULAS POR REAMOSTRAGEM  
DE IMPORTÂNCIA HÍBRIDA PARA  
RASTREAMENTO VISUAL**

**DIEGO ANTONIO LEGARDA CÓRDOBA**

**DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS  
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**FACULDADE DE TECNOLOGIA  
UNIVERSIDADE DE BRASÍLIA**

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

**PROPOSTA DE UM FILTRO DE PARTÍCULAS POR REAMOSTRAGEM  
DE IMPORTÂNCIA HÍBRIDA PARA  
RASTREAMENTO VISUAL**

**DIEGO ANTONIO LEGARDA CÓRDOBA**

**DISSERTAÇÃO DE MESTRADO ACADÊMICO SUBMETIDA AO DEPARTAMENTO  
DE ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA DA UNI-  
VERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS  
PARA A OBTENÇÃO DO GRAU DE MESTRE EM SISTEMAS MECATRÔNICOS.**

**APROVADA POR:**

---

**Prof. Flávio de Barros Vidal, Dr., CIC/UnB  
(Orientador)**

---

**Prof. Carla Maria Chagas e Cavalcante Koike, Dr., CIC/UnB  
Co-Orientador**

---

**Prof. Alexandre Zaghetto, DSc., CIC/UnB  
Membro Externo**

---

**Prof. José Maurício Santos Torres da Motta, Ph.D., ENM/UnB  
Membro Interno**

**BRASÍLIA, 01 DE MARÇO DE 2013.**

## FICHA CATALOGRÁFICA

LEGARDA CÓRDOBA, DIEGO ANTÔNIO  
PROPOSTA DE UM FILTRO DE PARTÍCULAS POR REAMOSTRAGEM DE IMPORTÂNCIA HÍBRIDA PARA RASTREAMENTO VISUAL [Distrito Federal] 2013.  
xvi, 89p., 210 x 297 mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2013).  
Tese de Mestrado – Universidade de Brasília, Faculdade de Tecnologia.  
Departamento de Engenharia Mecânica

1. Rastreamento Visual	2. Filtro Partículas
3. Robótica	4. Reamostragem Híbrida
I. ENM/FT/UnB	II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

CÓRDOBA, D. L. A. (2013). PROPOSTA DE UM FILTRO DE PARTÍCULAS POR REAMOSTRAGEM DE IMPORTÂNCIA HÍBRIDA PARA RASTREAMENTO VISUAL, Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação ENM.DM-57/2013, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 89p.

## CESSÃO DE DIREITOS

AUTOR: Diego Antonio Legarda Córdoba

TÍTULO: PROPOSTA DE UM FILTRO DE PARTÍCULAS POR REAMOSTRAGEM DE IMPORTÂNCIA HÍBRIDA PARA RASTREAMENTO VISUAL.

GRAU: Mestre                      ANO: 2013

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

---

Diego Antonio Legarda Córdoba  
Departamento de Eng. Mecânica (ENM) - FT  
Universidade de Brasília (UnB)  
Campus Darcy Ribeiro  
CEP 70919-970 - Brasília - DF - Brasil

*Dedicado a Esperanza Córdoba e a  
mulher do meu coração Andrea.*

## **AGRADECIMENTOS**

*A Deus com a sua infinita sabedoria me ensinou o caminho certo;*

*A meus pais Esperanza Córdoba e Floriberto Legarda pela confiança e apoio, a minha esposa Andrea Pantoja, pela paciência e compreensão, a minhas irmãs pelo constante apoio;*

*Ao professor Flavio Vidal, pelos constantes conselhos, confiança e orientação no desenvolvimento do projeto;*

*Á professora Karla Koike, pela ajuda e direcionamento;*

*A meus colegas do Graco pelas amostras de apoio e contribuições;*

*Ao meus amigos pelo ânimo nos momentos difíceis ;*

*Á CAPES pela bolsa de estudo concedida.*

## **RESUMO**

### **PROPOSTA DE UM FILTRO DE PARTÍCULAS POR REAMOSTRAGEM DE IMPORTÂNCIA HÍBRIDA PARA RASTREAMENTO VISUAL**

**Autor: Diego Antonio Legarda Córdoba**

**Orientador: Prof. Flávio de Barros Vidal, Dr., CIC/UnB**

**Programa de Pós-graduação em Sistemas Mecatrônicos**

**Brasília, Março de 2013**

O rastreamento mediante visão computacional constitui um campo de ampla pesquisa tendo em conta todas as aplicações possíveis. Para conseguir um bom desempenho do sistemas de rastreamento, propõem-se algoritmos estocásticos que lidam com problemas do ruído no sistema e na aquisição das observações do entorno. Dentre os métodos estocásticos encontra-se o filtro de partículas. Este filtro tem bom desempenho lidando com não-gaussianos, não-lineares, e multi-módulo problemas; quando os objetos rastreados não apresentam oclusões. Neste trabalho propõe-se um filtro de partículas com comportamento inercial que permita obter um comportamento robusto do rastreamento nos casos quando uma oclusão ocorrer mantendo o comportamento dinâmico do objeto, ação alcançada com a estimação da variação da posição ao longo dos frames anteriores à oclusão. Além de tratar as oclusões, o método exibe um ótimo desempenho em sequências com mudanças de iluminação, modificações bruscas de trajetória, com fundo variável e aqueles na qual existem mais de um alvo a rastrear. Os resultados obtidos demonstram que o método de rastreamento consegue um melhor desenvolvimento que o filtro padrão nas situações que precisam de um modelo do movimento do objeto a ser rastreado como são as sequências descritas.



## **ABSTRACT**

### **SAMPLING IMPORTANCE HYBRID PARTICLE FILTER FOR VISUAL TRACKING**

**Author: Diego Antonio Legarda Córdoba**

**Supervisor: Prof. Flávio de Barros Vidal, Dr., CIC/UnB**

**Programa de Pós-graduação em Sistemas Mecatrônicos**

Tracking through computer vision is a field of research extends taking into account all the possible applications. To achieve a good performance of tracking systems, stochastic algorithms are proposed to deal with problems of noise in the system and in the acquisition of the observations of the environment. Among stochastic methods is the particulate filter. dealing with the non-Gaussian, non-linear and multi-module problems. when the tracked objects have no occlusions, in addition to allow tracking of multiple targets. This paper proposes a particle filter with inertial behavior which results in a robust tracking behavior in cases when an occlusion occurs keeping the dynamic behavior of the object, achieved with the estimation of the variation in position over the frames prior to occlusion. Besides dealing with occlusions, the method displays a great performance sequences with changes in illumination, abrupt changes of trajectory, with variable background and those in which there are more than one target track. The results demonstrate that the tracking method achieves a better performance than the standard filter in situations that require a model of the movement of the object to be tracked as the sequences are described.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>1</b>
1.1	ORGANIZAÇÃO DA DISSERTAÇÃO .....	2
<b>2</b>	<b>RASTREAMENTO VISUAL DE OBJETOS</b> .....	<b>3</b>
2.1	CONCEITOS E DEFINIÇÕES BÁSICAS .....	3
2.2	REPRESENTAÇÃO DOS OBJETOS A SEREM RASTREADOS .....	4
2.3	CARACTERÍSTICAS COMUNS PARA RASTREAMENTO .....	5
2.3.1	COR .....	5
2.3.2	BORDAS .....	6
2.3.3	TEXTURA .....	6
2.3.4	CARACTERÍSTICAS ESPAÇO TEMPORAIS. ....	6
2.3.5	FUSÃO POR MÚLTIPLAS CARACTERÍSTICAS. ....	8
2.4	RASTREAMENTO POR CORRESPONDÊNCIA DE REGIÕES .....	9
2.4.1	FUNÇÕES DE CUSTO .....	10
2.5	RASTREAMENTO POR SEGMENTAÇÃO .....	10
<b>3</b>	<b>PRINCÍPIOS DE FILTRAGEM ESTOCÁSTICA</b> .....	<b>13</b>
3.1	FUNDAMENTOS BÁSICOS DO PROCESSO ESTOCÁSTICO .....	13
3.1.1	VARIÁVEL ALEATÓRIA .....	13
3.1.2	FUNÇÃO DENSIDADE PROBABILIDADE .....	13
3.1.3	REGULARIDADE ESTATÍSTICA .....	14
3.1.4	PROBABILIDADE CONDICIONAL .....	14
3.1.5	PROBABILIDADE TOTAL .....	14
3.1.6	REGRA DE BAYES .....	15
3.2	RASTREAMENTO UTILIZANDO FILTRAGEM ESTOCÁSTICA .....	15
3.2.1	REGRA DE BAYES PARA RASTREAMENTO .....	16
3.3	FILTRO DE PARTÍCULAS .....	17
3.3.1	AMOSTRAGEM SEQUENCIAL POR IMPORTÂNCIA .....	18
3.3.2	REAMOSTRAGEM .....	19
3.3.3	ALGORITMO DE AMOSTRAGEM E REAMOSTRAGEM POR IMPOR- TÂNCIA - SIR .....	20
3.4	FILTRO DE PARTÍCULAS BASEADO NO MÉTODO DE MONTE CARLO VIA CADEIAS DE MARKOV .....	21
3.4.1	ALGORITMO DE METROPOLIS-HASSTING .....	21
3.4.2	ESTIMAÇÃO .....	22
3.5	DISTÂNCIA BHATTACHARYYA .....	22

<b>4</b>	<b>METODOLOGIA PROPOSTA .....</b>	<b>24</b>
4.1	FLUXOGRAMA DO ALGORITMO PROPOSTO .....	24
4.2	MODELO E DISTRIBUIÇÃO DE CORES .....	24
4.2.1	MODELO HSV .....	24
4.3	ATRIBUIÇÃO DE PESOS .....	26
4.4	MODELO DE ATUALIZAÇÃO DO HISTOGRAMA .....	28
4.5	FILTRO DE PARTÍCULAS POR AMOSTRAGEM DE IMPORTÂNCIA HÍBRIDA ....	29
4.5.1	ESTRATÉGIA DE REAMOSTRAGEM DAS PARTÍCULAS .....	30
4.5.2	ALGORITMO METROPOLIS-HASTING .....	30
4.5.3	DISTRIBUIÇÃO A POSTERIORI .....	32
4.6	PROCESSO DE ATUALIZAÇÃO DE ESTADOS .....	34
4.6.1	AMOSTRAGEM.....	34
4.6.2	REAMOSTRAGEM.....	35
4.7	DETECÇÃO E TRATAMENTO DE OCLUSÕES .....	35
4.8	PSEUDOCÓDIGO DA METODOLOGIA PROPOSTA .....	36
4.9	ANÁLISE DA COMPLEXIDADE COMPUTACIONAL DO ALGORITMO PROPOSTO	38
<b>5</b>	<b>RESULTADOS DA IMPLEMENTAÇÃO .....</b>	<b>41</b>
5.1	RASTREAMENTO SIMPLES SEM OCLUSÃO.....	42
5.2	RASTREAMENTO COM VARIAÇÃO DE ESCALA ESPACIAL .....	43
5.3	RASTREAMENTO COM VARIAÇÃO DE TRAJETÓRIA E VELOCIDADE DO OB- JETO .....	44
5.4	RASTREAMENTO COM VARIAÇÃO DE <i>background</i> .....	44
5.5	RASTREAMENTO COM VARIAÇÃO DE ILUMINAÇÃO .....	46
5.6	RASTREAMENTO COM OCLUSÕES PARCIAIS E TOTAIS .....	48
5.7	RASTREAMENTO PARA MÚLTIPLOS ALVOS .....	54
5.8	COMPARATIVO ENTRE ALGORITMOS DE RASTREAMENTO E A METODO- LOGIA PROPOSTA .....	56
5.8.1	RASTREAMENTO SIMPLES SEM OCLUSÃO .....	58
5.8.2	RASTREAMENTO COM VARIAÇÃO DE ESCALA ESPACIAL .....	61
5.8.3	RASTREAMENTO COM VARIAÇÃO DE TRAJETÓRIA E VELOCIDADE DO OBJETO .....	62
5.8.4	RASTREAMENTO COM VARIAÇÃO DE <i>background</i> .....	63
5.8.5	RASTREAMENTO COM OCLUSÃO TOTAL .....	63
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS .....</b>	<b>70</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>72</b>
	<b>ANEXOS .....</b>	<b>76</b>
<b>A</b>	<b>ANÁLISE DA COMPLEXIDADE COMPUTACIONAL .....</b>	<b>77</b>

A.1 METODOLOGIA EMPREGADA .....	77
A.2 ANALISE DO FILTRO SIR(SAMPLING IMPORTANCE RESAMPLING) .....	77
A.3 ANALISE REAMOSTRAGEM .....	78

## LISTA DE FIGURAS

2.1	Exemplo de possíveis representações de Objetos a serem rastreados: (a) Centróide, (b) Multi-pontos, (c) Forma rectangular, (d) Forma elíptica, (e) Formas articuladas, (f) Esqueleto articulado, (g) Contorno de pontos, (h) Contorno contínuo e (i) Silhueta. [1].....	5
2.2	Método de rastreamento de bordas proposto por Gavrilin [2]: (a) Imagem Original (b) Modelo (c) Bordas da Imagem (d) Transformada de Distância. ....	7
2.3	Representação do método de segmentação de texturas, os pontos pretos e brancos representam 1 e 0 de acordo com o limiar de semelhança entre os pixels vizinhos .....	8
2.4	Exemplo de três regiões Ars [3] .....	9
2.5	Exemplo do processo de sobre-segmentação: (a)Imagem original e (b)Imagem sobre-segmentada baseada na similaridade de cores [4] .....	12
4.1	Fluxograma do algoritmo proposto.....	25
4.2	Modelo de Cores HSV (retirado de [5])......	26
4.3	Histograma H-S: a) Objeto a ser rastreado (envolvido pelo retângulo branco); b) Histograma de matiz e saturação do objeto.....	27
4.4	(a)Imagem original dividida em áreas iguais. O quadrado preto representa a área a ser comparada. (b)Valores da distância Bhattacharyya para cada região em comparação com o padrão de comparação. (c)Valores dos pesos das áreas da imagem. ....	29
4.5	Gráfica do numero de partículas e o numero de vezes que é empregado o processo de reamostragem.....	31
4.6	Exemplo conceitual de rastreamento de um objeto com movimento senoidal com diferentes amostragens. a) Amostragem padrão. b) Amostragem com Algoritmo Metropolis- Hasting. C) Amostragem sequencial por importância Híbrida com Algoritmo Metropolis-Hasting. ....	36
5.1	Resultado do rastreamento simples sem oclusão.....	42
5.2	Sequência de rastreamento sem oclusão. Posições Normalizadas Horizontal (a) e Vertical (b) descritas pelo <i>Ground truth</i> em relação à metodologia proposta do <a href="#">FPAH</a> . Erro da posição Horizontal (c) e Vertical (d).....	43
5.3	Sequência com mudança de escala espacial.....	44
5.4	Sequência com mudança de escala espacial. Posições Normalizadas Horizontal(a) e Vertical (b) descritas pelo <i>Ground truth</i> e pelo <a href="#">FPAH</a> . Erro da posição horizontal (c) e Vertical(d).....	45
5.5	Sequência com mudanças rápidas de trajetória e velocidade. ....	46

5.6	Sequência com mudança de trajetória. Posições Normalizadas Horizontal (a) e Vertical (b) descritas pelo <i>Ground truth</i> e pelo <b>FPAH</b> . Erro da posição Horizontal (c) e Vertical (d).....	47
5.7	Sequência com variação do <i>background</i> . .....	48
5.8	Sequência com variação do <i>background</i> . Posições Normalizadas Horizontal (a) e Vertical (b) descritas pelo <i>Ground truth</i> e pelo <b>FPAH</b> . Erro da posição Horizontal (c) e Vertical (d).....	49
5.9	Sequência com mudanças de iluminação. ....	50
5.10	Sequência com mudanças de iluminação. Posições Normalizadas Horizontal (a) e Vertical (b) descritas pelo <i>Ground truth</i> e pelo <b>FPAH</b> . Erro da posição Horizontal (c) e Vertical (d).....	51
5.11	Sequência com oclusão parcial de uma pessoa. ....	52
5.12	Sequência como oclusão parcial. Posições Normalizadas Horizontal (a) e Vertical (b) descritas pelo <i>Ground truth</i> e pelo <b>FPAH</b> . Erro da posição Horizontal (c) e Vertical (d).....	53
5.13	Sequência com oclusão total. ....	54
5.14	Sequência com oclusão total. Posições Normalizadas Horizontal (a) e Vertical (b) descritas pelo <i>Ground truth</i> e pelo <b>FPAH</b> . Erro da posição Horizontal (c) e Vertical (d).....	55
5.15	Sequência em ambiente externo com oclusão total.....	56
5.16	Sequência com oclusão <i>outdoor</i> . Posições Normalizadas Horizontal (a) e Vertical (b) descritas pelo <i>Ground truth</i> e pelo <b>FPAH</b> . Erro da posição Horizontal (c) e Vertical (d).....	57
5.17	Sequência de duas pessoas caminhando por um corredor com oclusão total e mudanças de iluminação.....	58
5.18	Sequência de rastreamento para múltiplos alvos. Posições Normalizadas horizontal (a) e vertical (b) descritas pelo <i>ground truth</i> para a pessoa A <b>GT-A</b> e para a pessoa B <b>GT-B</b> e pelo <b>FPAH</b> estimado para pessoa <b>FPAH-A</b> e pessoa B <b>FPAH-B</b> . Erro da estimação da posição para pessoas <b>A</b> , <b>B</b> nas orientações Horizontal(c) e Vertical(d).....	59
5.19	Desempenho dos algoritmos sequência sem oclusão: <b>WM</b> , <b>SIR</b> e <b>FPAH</b> . ....	60
5.20	Sequência sem oclusão. <i>Ground Truth</i> e estimação da posição dos algoritmos <b>WM</b> , <b>SIR</b> e <b>FPAH</b> orientação Horizontal(a) e Vertical(b). ....	60
5.21	Sequência com variação de escala espacial: <b>WM</b> , <b>SIR</b> e <b>FPAH</b> . ....	61
5.22	Sequência com variação de escala espacial. <i>Ground Truth</i> e estimação da posição dos algoritmos <b>WM</b> , <b>SIR</b> e <b>FPAH</b> nas direções Horizontal(a) e Vertical(b).....	62
5.23	Sequência com variação de trajetória e velocidade do objeto: <b>WM</b> , <b>SIR</b> e <b>FPAH</b> . ....	62

5.24	Sequência com variação de trajetória e velocidade do objeto. <i>Ground Truth</i> e posição dos algoritmos <b>WM</b> , <b>SIR</b> e <b>FPAH</b> nas direções Horizontal(a) e Vertical(b). .....	63
5.25	Sequência com variação de <i>background</i> : <b>WM</b> , <b>SIR</b> e <b>FPAH</b> . .....	64
5.26	Sequência com variação de <i>background</i> . <i>Ground Truth</i> e estimação da posição dos algoritmos <b>WM</b> , <b>SIR</b> e <b>FPAH</b> orientação Horizontal(a) e Vertical(b). ..	64
5.27	Sequência com oclusão total ( <i>indoor</i> ): <b>WM</b> , <b>SIR</b> e <b>FPAH</b> . .....	65
5.28	Sequência com oclusão total ( <i>indoor</i> ): <i>Ground Truth</i> e estimação da posição dos algoritmos <b>WM</b> , <b>SIR</b> e <b>FPAH</b> orientação Horizontal(a) e Vertical(b). .....	66
5.29	Sequência <i>outdoor</i> com oclusão total: <b>WM</b> , <b>SIR</b> e <b>FPAH</b> . .....	66
5.30	Sequência com oclusão <i>Ground Truth</i> e estimação da posição dos algoritmos <b>WM</b> , <b>SIR</b> e <b>FPAH</b> orientação Horizontal(a) e Vertical(b). .....	67

## LISTA DE TABELAS

5.1	Avaliação do erro médio quadrático e desvio padrão para os algoritmos avaliados em relação <i>ground truth</i> . .....	68
5.2	Dados da porcentagem de verdadeiros positivos em relação ao número de <i>frames</i> considerando de sobreposição dos algoritmos avaliados e o <i>ground truth</i> . .....	69

## LISTA DE SÍMBOLOS

$P(A)$	Probabilidade do evento $A$
$f_k(n)$	Frequência relativa do evento $k$
$N_k(n)$	Número de ocorrências do evento $K$
$\mathbf{X}_k$	Vetor de estados do sistema no instante $k$
$w_k$	Ruído de processo no instante de tempo $k$
$v_k$	Ruído de medição no instante de tempo $k$
$E(\mathbf{X}_k)$	Valor Esperado do vetor de estados $\mathbf{X}$ no instante de tempo $k$
$\mathbf{Y}_t$	Vetor de medições no instante de tempo $t$
$N(0, \sigma)$	Distribuição normal com media 0 e desvio padrão $\sigma$
$Var(w_k^*i)$	Variança dos pesos das partículas no instante $k$
$N_{eff}$	Tamanho efetivo de amostragem
$d_{H^o-H^i}$	Distância Bathacharyya entre os histogramas $H^o$ e $H^i$
$w_i$	Peso da partícula $i$ -ésima
$H_k^O$	Histograma do objeto no instante $k$
$H_k^{MP}$	Histograma da melhor partícula no instante $k$
$H_k^{\sum w_i \cdot X}$	Histograma ponderado das partículas no instante $k$
$N_f$	Número de frames onde o alvo está presente
$A(GT, ST)$	Relação entre a sobreposição da área do <i>ground truth</i> e a área de saída do sistema
<b>RGB</b>	Red Blue Green
<b>HSV</b>	Hue Saturation Value
<b>LBP</b>	Local Binary Patterns
<b>MB-ELBP</b>	Block Multiscale Enhanced Local Binary Patterns
<b>EBIM</b>	Enhanced Biologically Inspired Model
<b>Ars</b>	Attentional regions
<b>SIR</b>	Sampling Importance Resampling
<b>MB</b>	Medida Bathacharyya
<b>FPAD</b>	Filtro de Partículas com Amostragem Dupla
<b>BoBot</b>	Bonn Benchmark on tracking
<b>WM</b>	Windows Matching

# 1 INTRODUÇÃO

O objetivo do rastreamento visual é detectar, reconhecer e determinar a posição de um alvo e acompanhar sua trajetória a partir de sequências de imagens. Para detectar e reconhecer o objeto de interesse, este precisa ser caracterizado de maneira tal que possa se destacar dos demais objetos ao seu redor. Em seus primórdios, o rastreamento visual era realizado com a utilização da imagem capturada por um dispositivo de aquisição (tipicamente, uma câmera analógica em conjunto com uma placa conversora) e a ação de rastreamento visual dependia completamente dos olhos humanos. Com o desenvolvimento da tecnologia, e principalmente da acessibilidade ao hardware especializado em processamento de imagens digitais, os sistemas de rastreamento visual automático tornaram-se um interessante e amplo ramo de pesquisa.

O domínio de aplicação para as técnicas de rastreamento visual foi evoluindo gradativamente conforme a tecnologia das câmeras e computadores faziam seus próprios avanços. Paralelamente, a crescente demanda por análises automatizadas de vídeo têm gerado um grande interesse em algoritmos de rastreamento visual. Este desenvolvimento permite múltiplos campos de aplicações como: monitoramento de tráfego [6], soldagem industrial [7], automação industrial [8], monitoramento e vigilância [9] [10], navegação de veículos [11], segurança em condução de veículos [12], entre diversas outras aplicações em desenvolvimento.

O rastreamento visual pode ser o complemento ideal para aplicações relacionadas com a utilização de robôs em diferentes campos de atuação. Um exemplo muito comum envolvendo robótica e técnicas de rastreamento visual é o controle servovisual (do inglês *Visual Servoing*) em robôs manipuladores, onde emprega-se o rastreamento para acompanhar o objeto que é manipulado pelo robô [13][14]. Para alcançar um desempenho satisfatório do controle dos movimentos do robô, o algoritmo de rastreamento deve ser robusto e de baixa carga computacional para uma rápida resposta às mudanças ocorridas da posição do braço manipulador e do objeto manipulado. No caso de aplicações em robótica móvel, o rastreamento visual pode ser empregado principalmente na navegação e no auxílio do desvio de obstáculos [15]. Também pode ser utilizado para detecção e acompanhamento da trajetória do robô no ambiente fazendo o rastreamento a partir de uma câmera externa e no acompanhamento de pessoas em ambientes complexos.[16].

A obtenção da descrição do movimento de um objeto pela aquisição de imagens de vídeo implica em enfrentar alguns inconvenientes como: ruído gerado durante a captura da imagem pelo dispositivo de aquisição, variação da iluminação do ambiente, trajetórias complexas do objeto rastreado, tempo de análise e resposta do sistema, variações na forma e escala do alvo a ser rastreado, bem como oclusões parciais e totais do alvo, entre outros mais. Todas estes fatores fazem com que as metodologias de rastreamento visual tenham como preocu-

pação o aumento da robustez inerente a estes fatores, de forma que seja possível realizar o rastreamento com sucesso.

Outro benefício da evolução tecnológica das arquiteturas computacionais modernas, e do consequente aumento da capacidade de processamento de informações, é o aumento da frequência de aplicação de metodologias com características probabilísticas (como Filtro de Kalman, Filtro de Partículas, Cadeias de Markov, entre outras), que são de extrema utilidade para o aumento da robustez das técnicas de rastreamento visual, mas que exigem esforço computacional razoável.

Desta forma o presente trabalho propõe o uso de um filtro de partículas na elaboração de um algoritmo para rápida e precisa resposta diante de diferentes situações complicadas que podem ser encontradas durante o processo de rastreamento visual sob as condições mais comuns encontradas no ambiente de aplicação, a saber: mudanças de iluminação, variação de contorno do objeto, oclusões parciais e totais, . A metodologia proposta para a implementação do Filtro de Partículas foi definida a partir da incorporação de técnicas que permitissem a melhoria da robustez do filtro, mas sem comprometer seu desempenho, reduzindo significativamente a carga computacional utilizada em sua implementação e realização.

## 1.1 ORGANIZAÇÃO DA DISSERTAÇÃO

A organização da Dissertação foi definida como : 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 de acordo com o tipo de estrutura utilizada no seu desenvolvimento. O Capítulo 3 descreve as idéias básicas em sistemas probabilísticos básicos, necessárias para o entendimento das técnicas de filtragem estocástica utilizadas, bem como os princípios da utilização destas para o rastreamento. No Capítulo 4 é apresentada a descrição geral do sistema de rastreamento proposto a partir da descrição modular dos principais componentes do algoritmo proposto. Neste capítulo também são abordadas análises da robustez e complexidade computacional da metodologia proposta. O Capítulo 5 apresenta a proposta de algoritmo de rastreamento e seus resultados em diferentes sequências de imagens, sendo estes resultados comparados com outros dois métodos de rastreamento (Window-Matching e uma versão básica de um Filtro de Partículas). Os resultados obtidos pelas metodologias testadas são comparados aos de um *Benchmark* específico para análise de robustez e precisão de técnicas de rastreamento visual. 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 DE OBJETOS

No decorrer deste capítulo serão apresentadas as principais metodologias que serviram de inspiração e motivação para o desenvolvimento das técnicas de rastreamento visual desta dissertação. Entretanto, faz-se de suma importância ressaltar que as técnicas aqui descritas não esgotam em termos descritivos a imensa variedade e possibilidades de técnicas que podem ser encontradas na literatura. Existem vários trabalhos que tentam, de forma ambiciosa, uma abordagem descritiva das técnicas comumente utilizadas. Desta forma, utilizando como referência estes trabalhos, propõe-se aqui uma breve adequação de vocabulário e descrição de técnicas que foram utilizadas como referência ou comparadas com os resultados obtidos pela metodologia proposta e desenvolvida neste trabalho.

### 2.1 CONCEITOS E DEFINIÇÕES BÁSICAS

As metodologias de rastreamento visual ainda são assunto de elevada relevância para a comunidade científica de visão computacional, principalmente devido às grandes quantidades de aplicações como: sistemas de vigilância, video-monitoramento, interações homem-máquina e veículos autônomos, etc. O rastreamento visual de objetos tem sido alvo de grandes esforços nas últimas décadas, em que o desafio pela busca do desenvolvimento de metodologias robustas referentes aos aspectos relacionados principalmente à variação de aparência do objeto (alvo) e dos fatores externos ao objeto rastreado que dificultam todo o processo de rastreamento.

O rastreamento visual de objetos tem como principal objetivo seguir um objeto durante sua trajetória desenvolvida por todos os quadros de uma seqüência de imagens, mantendo uma estimativa correta sobre sua trajetória e tamanho ao longo do tempo. Os principais fatores externos que dificultam a correta realização deste tipo de metodologia ocorrem em situações quando há diversas alterações no ambiente escolhido de aplicação e/ou elevado grau de semelhança entre o ambiente e o alvo. Além destas situações, ainda é possível se deparar com problemas de oclusões que ocorrem quando outro objeto (ou elemento que compõe o ambiente) oculta parcial ou totalmente o alvo a ser rastreado. Outro problema que pode ser encontrado é o efeito de *feature drift*. O *feature drift* ocorre quando ao modelo do objeto rastreado, são incorporadas informações além do contorno do objeto, como parte da região de interesse a ser rastreada. Com o tempo, o objeto a ser rastreado se perde, dificultando a realização do rastreamento visual. Adicionalmente todos estes fatores ainda podem ser agravados pelas mudanças de iluminação, erros de quantização de vídeo, entre outros fatores que podem ser encontrados no ambiente de aplicação deste tipos de técnicas.

De acordo com [17], as metodologias desenvolvidas para o rastreamento visual de ob-

jetos em seqüências de imagens devem possuir as seguintes características básicas para que alcance sucesso em sua realização, a saber:

- **Robustez** - O método tem que ser capaz de seguir o alvo em diversas situações como fundo da imagem difusa (i.e., *cluttered environment*), mudanças de iluminação, oclusões e movimentos complexos do objeto durante sua trajetória.
- **Adaptabilidade** - Mesmo com as variações do fundo da imagem, o alvo também pode realizar mudanças bruscas de trajetória que podem alterar a sua forma de contorno durante o rastreamento (por exemplo, objetos rígidos assimétricos realizando uma rotação). O algoritmo tem que ser capaz de adaptar-se à nova aparência do objeto na cena.
- **Processamento em tempo real** - É desejável que aplicações de visão computacional sejam capazes de realizar as atividades designadas respeitando o critério de tempo real de execução. Neste caso, a carga computacional de processamento empregada deve ser tal que permita à metodologia proposta executar todos os procedimentos com tempo de processamento menor do que taxa de aquisição do sistema de captura de imagens.

## 2.2 REPRESENTAÇÃO DOS OBJETOS A SEREM RASTREADOS

Durante o rastreamento, um objeto pode ser representado de acordo com a finalidade desejada para a aplicação, possibilitando análises posteriores. De acordo com [1], dentre as diferentes formas de representar o objeto que se deseja rastrear, podemos citar:

- **Pontos** - Geralmente um único ponto é utilizado para representar o centróide do alvo. Quando se tem vários pontos, estes podem abranger outras partes do objeto em questão.
- **Formas geométricas primitivas**- Dentre as formas geométricas básicas, pode se utilizar uma elipse (ou um retângulo) que delimita o objeto a ser rastreado. Esta representação é ideal para alvos rígidos, mas também pode ser utilizada em objetos não rígidos.
- **Silhueta/Contorno do objeto** - O contorno do objeto é formado pelas bordas e a região interna do contorno é a silhueta. Este tipo de representação é muito útil para alvos com formas complexas, como por exemplo, formas não rígidas.
- **Formas articuladas** - Este tipo de representação decompõe a forma do objeto como um conjunto de peças articuladas, conectadas por meio de juntas. Neste caso, estas reagem de acordo com o modelo cinemático do movimento.

A Figura 2.1, reproduzida de [1], apresenta exemplos de algumas possíveis representações para o objeto a ser rastreado. A escolha de qual forma será utilizada afetará diretamente

a capacidade da metodologia escolhida de realizar o rastreamento com sucesso. Por exemplo, objetos não-rígidos dificultam o rastreamento quando se é escolhido uma forma de representação fixa (como um retângulo).

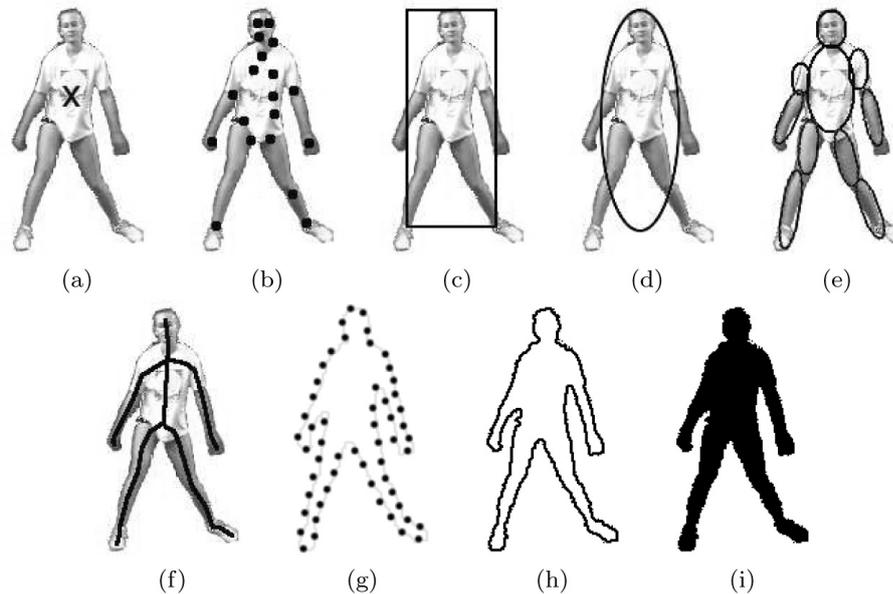


Figura 2.1: Exemplo de possíveis representações de Objetos a serem rastreados: (a) Centróide, (b) Multi-pontos, (c) Forma rectangular, (d) Forma elíptica, (e) Formas articuladas, (f) Esqueleto articulado, (g) Contorno de pontos, (h) Contorno contínuo e (i) Silhueta. [1]

## 2.3 CARACTERÍSTICAS COMUNS PARA RASTREAMENTO

Algoritmos de rastreamento por visão computacional, em sua maioria, utilizam alguma característica própria e relevante do objeto a ser rastreado, de maneira tal, que permita sua distinção dos demais elementos constituintes da cena. Estas características são utilizadas como parâmetro principal de entrada nestes tipos de algoritmos com o intuito de permitir sua correspondência nas imagens subseqüentes. Inspirado no trabalho de [1], as subseções a seguir listam as principais características utilizadas em algoritmos de rastreamento visual.

### 2.3.1 Cor

A cor aparente de um objeto está delimitada por duas propriedades físicas: a *distribuição espectral de iluminação* e a *reflectância das superfícies iluminadas*[1]. Dependendo da aplicação, pode-se escolher o modelo de cor **RGB** que decompõe a cor como uma combinação de três cores: o vermelho (R), o verde (G) e o azul (B). A dificuldade que este modelo apresenta é sua inerente sensibilidade às mudanças naturais de cor para a percepção do olho humano. Outro modelo de cor também muito utilizado é o **HSV** (do inglês *Hue, Saturation and Value*), que representa uma determinada cor como uma associação de três componentes:

a Matiz (H) que corresponde a intensidade espectral de cor, Saturação (S) que representa o grau de pureza de uma cor e o Valor (V) que corresponde a intensidade luminosa (brilho). Este modelo é representado por um espaço de cor uniforme à percepção humana.

### **2.3.2 Bordas**

Geralmente, as bordas dos objetos nas imagens geram grandes variações de intensidade de brilho, o que permite a utilização desta informação como uma característica a ser rastreada em uma seqüência de imagens. Esta variação permite a implementação de técnicas baseadas no gradiente espacial destas regiões de borda. As bordas são pouco sensíveis às mudanças de iluminação se comparadas com métodos baseados na cor dos objetos. A título de exemplo, na Figura 2.2 é apresentada uma metodologia para rastreamento de pedestres proposta por Gavrilin [2] em que foi utilizado como informação principal o contorno (borda) do objeto a ser rastreado em conjunto com a classificação hierárquica baseada em modelo com a transformada de distância de Hausdorff [18].

### **2.3.3 Textura**

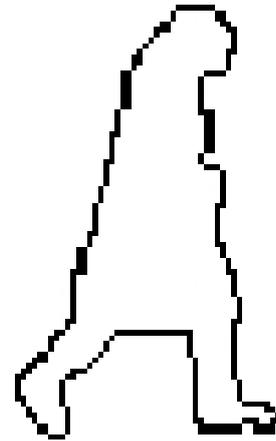
A textura permite definir a variação da intensidade luminosa de uma superfície, em que a partir desta característica gera-se um descritor composto por filtros sucessivos para definir as informações de suavidade e regularidade. Em Ojala *et al.* [19] é apresentado um algoritmo para segmentação de texturas em imagem a partir da amostragem circular de píxeis e média de histogramas normalizados (Figura 2.3 reproduzida de [19]). Estes descritores em escala de níveis de cinza foram denominados *Padrões Binários Locais* (do inglês *Local Binary Patterns*). Estes descritores gerados no processo de segmentação podem ser utilizados como entrada de um algoritmo de rastreamento visual em imagens.

### **2.3.4 Características espaço temporais.**

Ao se optar por descritores locais em seqüências de imagens, torna-se possível estabelecer a relação espaço-tempo do movimento descrito no domínio da imagem, bem como as variações de escala de forma independente. O trabalho apresentado em [20] propõe uma metodologia de avaliação para combinações de detectores e descritores, utilizando diferentes conjuntos de dados para se obter uma correlação adequada entre eles. Este tipo de abordagem permite, como apresentado por [21], a obtenção de um descritor com vetores tridimensionais para imagens. Este descritor pode ser obtido realizando a integração das informações provenientes de seus gradientes e de suas orientações.



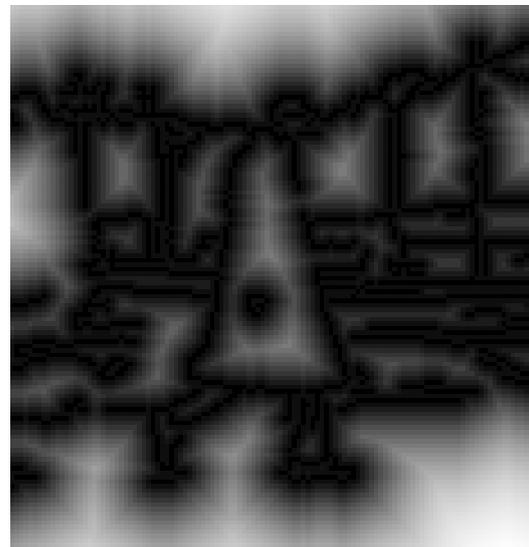
(a)



(b)



(c)



(d)

Figura 2.2: Método de rastreamento de bordas proposto por Gavrilu [2]: (a) Imagem Original (b) Modelo (c) Bordas da Imagem (d) Transformada de Distância.

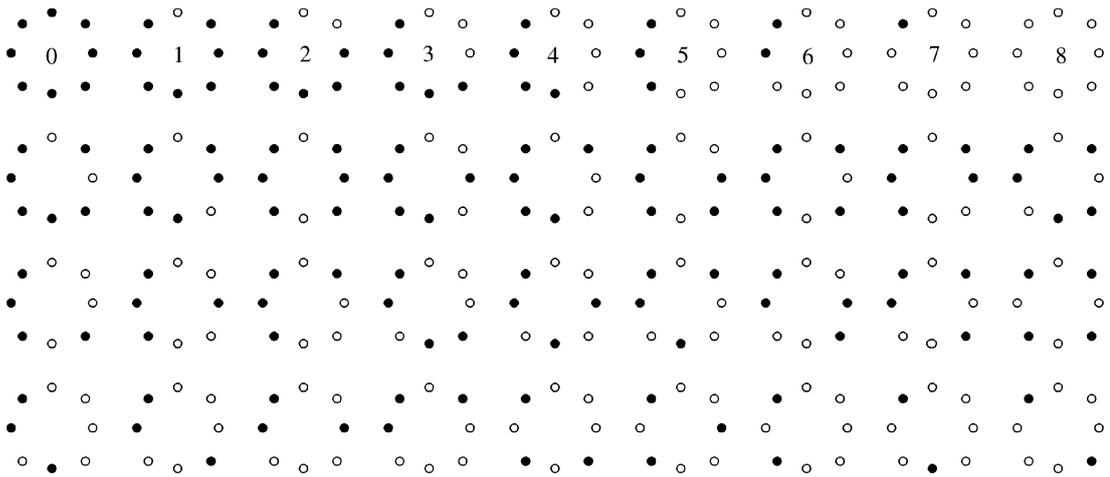


Figura 2.3: Representação do método de segmentação de texturas, os pontos pretos e brancos representam 1 e 0 de acordo com o limiar de semelhança entre os pixels vizinhos

### 2.3.5 Fusão por múltiplas características.

Dependendo do tipo de aplicação de rastreamento desejada, a utilização de métodos que realizam a incorporação (ou fusão) de diversas características do objeto a ser rastreado fornecem resultados relevantes. Neste tipo de abordagem pode-se fazer uso de características ou informações diversas, como: intensidade luminosa, bordas, cor, silhueta, etc.

Entre as diversas aplicações que utilizam esta combinação de características descritivas, podemos destacar o trabalho apresentado em [22]. Neste trabalho, foi utilizada uma representação por uma matriz de covariância das características discriminantes do alvo a ser rastreado a partir das informações provenientes da geometria de Riemann [23].

Trabalhos recentes inspiraram técnicas de rastreamento por fusão de múltiplas características a partir da forma como o ser humano consegue agir para reconhecer os objetos. Em uma tradução livre do termo em inglês *Enhanced Biologically Inspired Model* (EBIM), este tipo de metodologia foi denominada "Melhoria do Modelo por Inspiração Biológica". Foi a partir desta abordagem que, no trabalho descrito em [24], foram feitas inovações para melhorar o modelo dos objetos a serem rastreados, inspirado no córtex visual humano primário (encarregado do processamento de informação dos objetos estáticos e em movimento e que também auxilia no reconhecimento de padrões). Variantes desta modalidade de rastreamento podem ser encontradas no trabalho [3], em que é proposta a definição de Regiões de Atenção (do inglês *Attentional Regions* (Ars) tendo como base a dinâmica da percepção visual humana, no qual o processamento concentra-se na informação relevante e importante do sistema de rastreamento visual. A título de exemplo, a Figura 2.4, reproduzida de [3], ilustra uma abstração de possíveis descritores Ars que podem ser utilizados no quadro da Mona Lisa de Da Vinci.

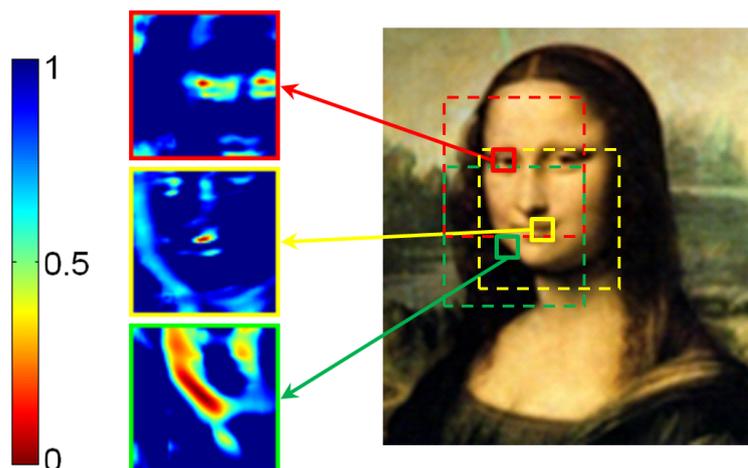


Figura 2.4: Exemplo de três regiões Ars [3]

## 2.4 RASTREAMENTO POR CORRESPONDÊNCIA DE REGIÕES

Esta metodologia de rastreamento deve ser utilizada para encontrar a correspondência de um ponto de interesse a ser rastreado em um quadro da imagem no instante de tempo  $t - 1$  no instante de tempo  $t$  a partir da utilização de uma *Função de custo*, como apresentado em [25]. Com esta abordagem, é possível extrapolar a idéia de correlação ponto-a-ponto da imagem para a correlação entre as regiões da imagem.

Entretanto, por se tratar de um procedimento basicamente determinístico de rastreamento, esta metodologia apresenta algumas restrições:

- **Suavidade espacial:** O objeto a ser rastreado não poderá realizar movimentos abruptos entre um quadro e outro na sequência de imagens, ou seja o movimento dos pixels tem que manter uma distancia curta de variação entre um frame e outro;
- **Máxima velocidade:** Define a velocidade limite superior em que pode se movimentar o objeto e limita a correspondência da vizinhança do objeto a ser rastreado;
- **Baixas variações de velocidade:** Assume-se que a direção e o módulo da velocidade não mudam de forma extrema;
- **Corpo Rígido:** Assume-se que a forma tridimensional do objeto permanece constante durante sua trajetória, ou seja, a distância entre dois pontos do objeto permanece constante ao longo do tempo.

Estas restrições não se aplicam somente para os métodos determinísticos e algumas delas podem apresentar-se em metodologias estocásticas, A proposta desenvolvida neste trabalho, não sofre de estas limitações devido a seu comportamento inercial que aumenta a sua velocidade de resposta diante mudanças rápidas do objeto rastreado.

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

$$f(J, J'(w, h)). \quad (2.1)$$

Sendo uma medida da disparidade (separação ou distância entre os elementos correspondentes), a procura da correspondência entre as regiões consistirá então na minimização (ou maximização) desta função de custo. A escolha de processos de minimização ou maximização dependerá exclusivamente do tipo da função de custo utilizada. No caso da utilização de funções de correlação será a maximização e para funções de similaridades faz-se uso da minimização.

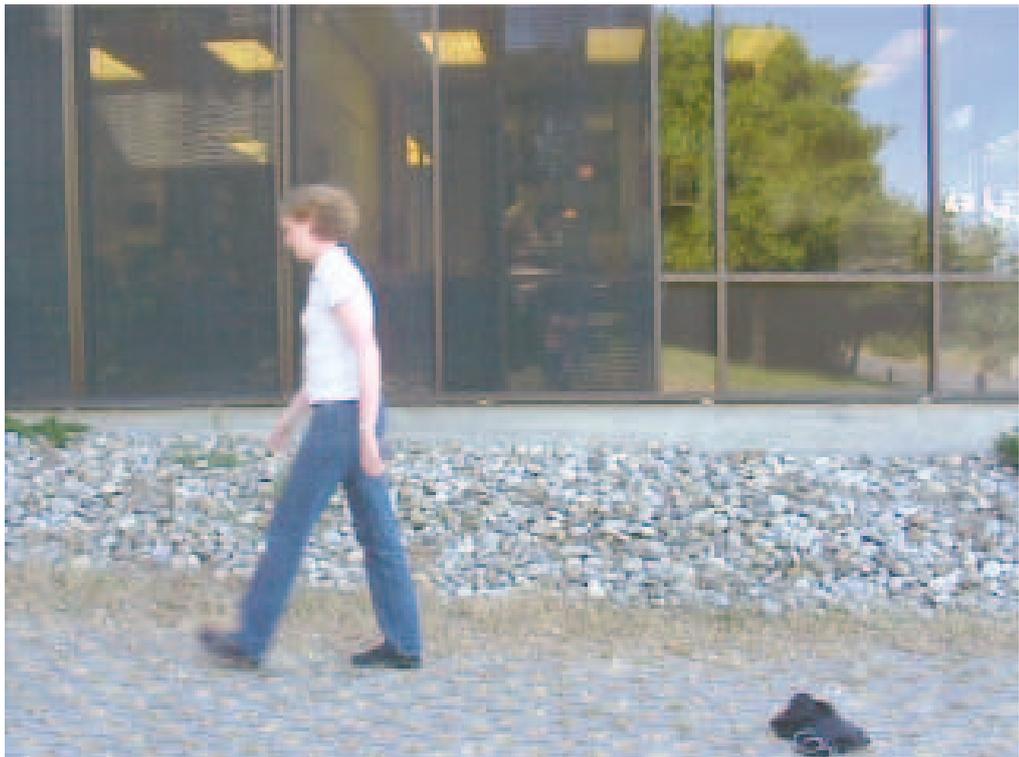
### 2.4.1 Funções de Custo

Em [26], são apresentadas *Funções de Custo* empregadas na correlação entre sinais unidimensionais (ex.: áudio) e bidimensionais (ex.: imagens). 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 que demandam procedimentos de minimização das funções de custo. Outras aplicações que utilizam funções de custo podem ser encontradas nos trabalhos de [27], [28], [29], [30], [31] e [32].

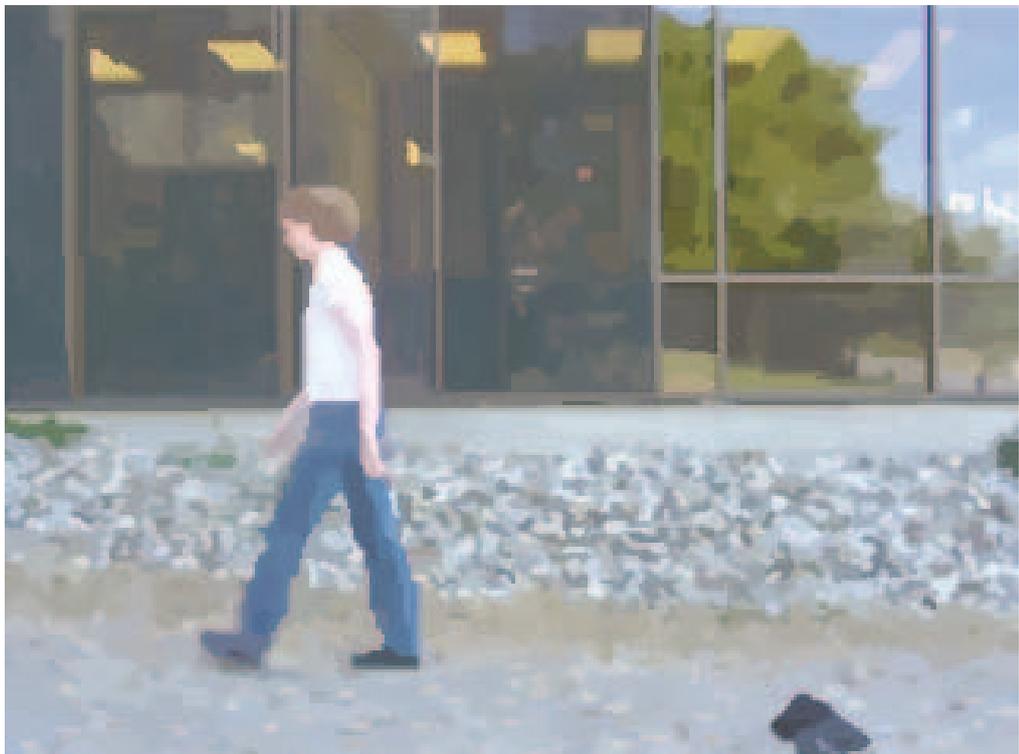
## 2.5 RASTREAMENTO POR SEGMENTAÇÃO

Em visão computacional, entende-se a segmentação como a divisão de uma imagem em partes que compartilham alguma característica em comum. Existem diferentes algoritmos e métodos de segmentação baseados em informações temporais e espaciais provenientes da imagem. Desta forma, se estas regiões podem ser detectadas ao longo da execução no tempo em uma seqüência de imagens, este processo pode ser considerado como uma metodologia de rastreamento baseada na segmentação de imagens. Todo processo de segmentação necessita da elaboração de algum tipo de descritor específico, que seja capaz de detectar a informação de interesse do objeto em que se deseja rastrear. A título de exemplo, é possível encontrar na literatura comum da área de visão computacional algoritmos de rastreamento baseados na segmentação das características descritas na Seção 2.3.

Algumas variações desta metodologia, em uma abordagem melhorada e mais eficiente, podem ser facilmente encontradas na literatura. O trabalho de Wang e Yagi [4] descreve um algoritmo de sobre-segmentação (Figura 2.5), em que se divide o objeto em sub-regiões para acelerar o processo de segmentação. Para tal, faz-se o uso de árvores com as informações de cor (no espaço RGB, por exemplo) e orientação espacial das regiões a serem segmentadas. A Figura 2.5-(b) descreve o resultado do processo de sobre-segmentação proposto por [4].



(a)



(b)

Figura 2.5: Exemplo do processo de sobre-segmentação: (a)Imagem original e (b)Imagem sobre-segmentada baseada na similaridade de cores [4]

### **3 PRINCÍPIOS DE FILTRAGEM ESTOCÁSTICA**

Neste capítulo, serão descritas as principais metodologias utilizadas como ferramental teórico na implementação de filtragem estocástica. Ressaltamos que não se tem o objetivo de esgotar o assunto, mas sim descrever os princípios básicos da utilização de filtros estocásticos para rastreamento visual.

#### **3.1 FUNDAMENTOS BÁSICOS DO PROCESSO ESTOCÁSTICO**

Ao estudar um fenômeno real, surge o problema relacionado à obtenção de modelos que se aproximem ao máximo do comportamento real do fenômeno em estudo. É comum fazer uso de modelos matemáticos simplificados devido à elevada complexidade em obter modelos fiéis ao fenômeno real que deseja estudar. Com este modelo matemático simplificado, busca-se uma quantificação do comportamento dos fenômenos em estudo, mas em diversas situações estes modelos matemáticos não conseguem refletir este comportamento, sendo necessário o emprego de métodos mais detalhistas, muitas vezes com elevado grau de complexidade. Uma destas modalidades de modelagem detalhista propõe uma solução probabilística de modelagem utilizando uma representação por processos estocásticos. Os processos estocásticos são representados por variáveis aleatórias e funções destas variáveis aleatórias.

Diferente dos métodos estatísticos amostrais, a resposta de um processo estocástico será uma função ou uma sequência de valores e não apenas um número [33]. Essa função em geral representa a densidade de probabilidade de um conjunto de resultados possíveis para uma determinada variável aleatória.

##### **3.1.1 Variável Aleatória**

Uma variável aleatória é uma variável cujo valor é igual ao resultado de um experimento aleatório, ou seja, representa todo o espaço amostral da experiência realizada. Dependendo do tipo de experimento, esta variável pode ser de natureza contínua ou discreta[33].

##### **3.1.2 Função Densidade Probabilidade**

Para uma variável aleatória contínua, a função densidade probabilidade descreve a probabilidade de uma variável aleatória assumir um determinado valor dentro do conjunto dado. Em outras palavras, a função densidade probabilidade é uma expressão matemática que representa o comportamento provável de um conjunto de resultados possíveis da variável ale-

atória [33].

### 3.1.3 Regularidade estatística

Um modelo matemático deve possuir a capacidade de definir o comportamento de um sistema ao longo do tempo. Para isso, observa-se a resposta de um experimento obtido depois de realizar as experiências um determinado número de vezes  $n$ , e o valor da resposta obtida tende a um valor médio do conjunto de resultados esperados. A este fenômeno confere-se o termo de *regularidade estatística*.

Ao se repetir muitas vezes o experimento, ou seja quando  $n$  tende ao infinito, têm-se

$$\lim_{n \rightarrow \infty} f_k(n) = p_k, \quad (3.1)$$

onde  $f_k$  representa a frequência de ocorrência do evento  $k$  como resultado da experiência,  $p_k$  é conhecido como a probabilidade do evento  $k$ .

### 3.1.4 Probabilidade condicional

Supondo que temos dois espaços amostrais  $A$  e  $B$ , e sabemos que eles estão relacionados entre si, de tal maneira que o conhecimento sobre o evento  $B$  nos dê uma informação sobre o evento  $A$ . Para modelar essa informação, empregamos a probabilidade condicional  $p(A|B)$ , que pode ser definida como:

$$p(B|A) = \frac{p(B \cap A)}{p(A)} \quad (3.2)$$

Esto quer dizer que para encontrar a probabilidade de  $A$ , tendo como certeza que o evento  $B$  ocorre, O espaço amostral  $S$ , ao qual pertencem  $A$  e  $B$ , se reduz ao espaço  $B$  com a expressão  $p(B)$ , além de depender da probabilidade de que eventos que pertencem ao espaço  $A$  e  $B$  ocorram, amostrada por la expressão  $p(B \cap A)$  para avaliar a ocorrência de  $A$ .

### 3.1.5 Probabilidade total

Seja  $S$  o espaço amostral de um dado experimento que foi dividido em  $p_i$  partes. Define-se  $A$  um evento contido no espaço amostral, então pode-se definir que:

$$A = (A \cap S) \quad (3.3)$$

e que:

$$S = \sum_i^n p_i. \quad (3.4)$$

Então a probabilidade de  $p(A)$  pode ser estendida como:

$$p(A) = P(A \cap S). \quad (3.5)$$

Substituindo  $S$  e aplicando a regra de distribuição tem-se:

$$p(A) = \sum_i^n p(A \cap p_i). \quad (3.6)$$

Manipulando a equação anterior, temos que:

$$p(A) = \sum_i^n p(A|p_i)p(p_i) \quad (3.7)$$

### 3.1.6 Regra de Bayes

A regra de Bayes considera a relação entre incertezas no conhecimento da ocorrência de um evento. De acordo com [34], a partir de dois eventos mutualmente dependentes  $A$  e  $B$ , pode-se escrever uma expressão para a probabilidade condicional seguindo a regra de Bayes como apresentado na Equação 3.8.

$$p(A|B) = \frac{p(B|A)P(A)}{p(B)}, \quad (3.8)$$

em que  $p(A|B)$  é a probabilidade *a posteriori*,  $p(B|A)$  a probabilidade de verossimilhança,  $p(A)$  e  $p(B)$  são as probabilidades *a priori* dos eventos. Com a regra de Bayes é possível conhecer o comportamento de  $A$ , tendo como base o comportamento individuais de  $A$  e  $B$  anteriores e sua probabilidade conjunta para obter uma estimação resultante.

## 3.2 RASTREAMENTO UTILIZANDO FILTRAGEM ESTOCÁSTICA

O processo de rastreamento utilizando filtragem estocástica é tratado como um problema de estimação. A idéia fundamental é obter a estimação precisa da posição atual do objeto a ser rastreado a partir das observações.

A partir desta prerrogativa, o objeto a ser rastreado é definido por uma variável de estado vetorial que permite caracterizar este objeto. No caso do rastreamento visual, esta variável pode ser um vetor que contenha as posições horizontais e verticais no domínio da imagem, bem como quaisquer outras informações como média do histograma normalizado, por exemplo.

Dessa forma, o rastreamento pode ser definido como uma sequência de estados  $\{X_t, t \in N\}$  sendo o modelo de espaço de estados definido pela Equação 3.9.

$$\begin{aligned} X_t &= f_t(\mathbf{X}_{t-1}, w_{t-1}) \\ Y_t &= h_t(\mathbf{X}_t, v_t) \end{aligned} \quad (3.9)$$

Onde  $f(\cdot)$  e  $h(\cdot)$  são geralmente definidas como funções não lineares,  $\mathbf{X}_t$  é o vetor de estado do objeto no instante de tempo  $t$ , e  $\mathbf{Y}_t$  representa o conjunto de observações do objeto no instante  $t$ .  $w$  representa a incerteza da variável de estado descrita por uma função de variável aleatória.  $v$  representa a incerteza nas observações, com a mesma natureza estocástica que  $w$ , sendo  $w$  e  $v$  independentes [35].

O objetivo do rastreamento é obter uma estimação com certo nível de incerteza do estado de  $\mathbf{X}$ , utilizando o estado anterior e as observações realizadas. A partir desta afirmação, a Equação 3.9 é escrita na forma probabilística como:

$$\begin{aligned} \mathbf{X}_t &\sim (\mathbf{X}_t | \mathbf{X}_{t-1}, w_{t-1}) \\ \mathbf{Y}_t &\sim (\mathbf{Y}_t | \mathbf{X}_t, v_t). \end{aligned} \quad (3.10)$$

De acordo com [36], o processo de estimação, partindo do estado anterior e das observações do objeto, pode ser resumido em duas etapas:

- **Predição:** Com o modelo da equação dinâmica e a função densidade de probabilidade de uma variável a ser estimada, calculada no instante de tempo  $t - 1$ , obtém-se a função densidade de probabilidade *a priori* do estado atual, ou seja  $p(\mathbf{X}_t | \mathbf{X}_{t-1})$ .
- **Atualização:** A partir das observações no instante de tempo  $t$  definidas por  $p(\mathbf{Y}_t | \mathbf{X}_t)$ , obtém-se a função densidade probabilidade *a posteriori*  $p(\mathbf{X}_t | \mathbf{Y}_t)$ .

Para ambas etapas, assume-se que as variáveis são mutuamente independentes e ainda faz-se necessária a hipótese de que é suficiente a utilização do estado imediatamente anterior para estimar o estado atual, ou seja:

$$p(\mathbf{X}_t | \mathbf{X}_1 \dots \mathbf{X}_{t-1}) = p(\mathbf{X}_t | \mathbf{X}_{t-1}). \quad (3.11)$$

A partir da consideração anterior, vale ressaltar que as observações dependem somente do estado atual, e portanto, cada observação é independente das demais já obtidas, conforme apresentado em [35]:

$$p(\mathbf{Y}^i, \mathbf{Y}^j \dots \mathbf{Y}^l | \mathbf{X}^i) = p(\mathbf{Y}^i | \mathbf{X}^i) P(\mathbf{Y}^j \dots \mathbf{Y}^l | \mathbf{X}^i) \quad (3.12)$$

### 3.2.1 Regra de Bayes para rastreamento

Para obter uma estimativa do estado  $\mathbf{X}_t$ , é preciso obter diferentes valores das observações  $\mathbf{Y}_t$  no instante  $t$ , requeridas para construir a função densidade de probabilidade

$p(\mathbf{X}_t|\mathbf{Y}_t)$ . Assuma-se que a função densidade de probabilidade inicial  $p(\mathbf{X}_0|\mathbf{Y}_0) = p(\mathbf{X}_0)$  ou função densidade de probabilidade *a priori* do vetor de estado. Suponha-se que é possível avaliar  $p(\mathbf{X}_{t-1}|\mathbf{Y}_{t-1})$  no tempo  $t - 1$ . O estado de predição empregando o modelo de estado dado por 3.9 e a Equação Chapman-Kolmogorov [36] pode ser escrito como:

$$p(\mathbf{X}_t|\mathbf{Y}_{t-1}) = \int p(\mathbf{X}_t|\mathbf{X}_{t-1})p(\mathbf{X}_{t-1}|\mathbf{Y}_{t-1})dX_{t-1} \quad (3.13)$$

Aqui, a equação tem em consideração as condições de independência definidas pelas equações 3.11 e 3.12, por tanto,  $p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{Y}_{t-1}) = p(\mathbf{X}_t|\mathbf{X}_{t-1})$ . O modelo de probabilístico de evolução de estados  $p(\mathbf{X}_t|\mathbf{X}_{t-1})$  é definido pelo modelo 3.9 e pelo conhecimento da evolução de  $w_{t-1}$ .

Agora sim ao tempo  $t$  e realizada uma medida  $\mathbf{Y}_t$  da observação do estado, esta pode ser empregada para atualizar o estado *a priori* com a regra de Bayes, como apresentado a seguir:

$$p(\mathbf{X}_t|\mathbf{Y}_t) = \frac{p(\mathbf{Y}_t|\mathbf{X}_t)p(\mathbf{X}_t|\mathbf{Y}_{t-1})}{p(\mathbf{Y}_t|\mathbf{Y}_{t-1})}. \quad (3.14)$$

onde a constante de normalização

$$p(\mathbf{Y}_t|\mathbf{Y}_{t-1}) = \int p(\mathbf{Y}_t|\mathbf{X}_t)p(\mathbf{X}_t|\mathbf{Y}_{t-1})dX_t. \quad (3.15)$$

depende da função de probabilidade  $p(\mathbf{Y}_t|\mathbf{X}_t)$  definido pelo modelo de observação 3.9 e a evolução do  $v$ . Na etapa de atualização 3.14, a observação  $\mathbf{Y}_k$  é utilizada para modificar a função densidade de probabilidade *a priori* e obter a função densidade de probabilidade *a posterior* requerida para o estado atual.

A partir de todas as premissas e equações anteriores o valor esperado de  $E_{p(\mathbf{X}_t|\mathbf{Y}_t)} \langle \mathbf{X}_t \rangle$  pode ser definido como:

$$E_{p(\mathbf{X}_t|\mathbf{Y}_t)} \langle \mathbf{X}_t \rangle = \int \mathbf{X}_t p(\mathbf{X}_t|\mathbf{Y}_t) dX_t \quad (3.16)$$

### 3.3 FILTRO DE PARTÍCULAS

Para obtenção da função densidade de probabilidade *a posteriori*, o filtro de partículas é considerado uma boa alternativa, devido a sua resposta a sistemas não-lineares multimodais [37], assim como pelo fato de ser baseado no método de Monte Carlo[38]. Atualmente existe uma vasta literatura a respeito das diversas formas de implementação para o Filtro de Partículas. Neste trabalho, é utilizada a versão inspirada na Amostragem Sequencial por Importância (do inglês *Sequential importance resampling (SIR)*) e na seção a seguir são apresentadas as principais características desta implementação.

### 3.3.1 Amostragem Sequencial por Importância

A ideia principal deste método é conseguir uma representação da função de densidade de probabilidade *a posteriori* com um conjunto de amostras aleatórias com pesos associados.

Seja  $\{\mathbf{X}_k^i, w_k^i\}_{i=1}^{N_s}$  um estado aleatório, sobre um estado  $\mathbf{X}$  que caracteriza a função de densidade de probabilidade *a posteriori*  $p(\mathbf{X}_k|\mathbf{Y}_k)$ , onde  $(\mathbf{X}_k^i, i = 0, \dots, N_s)$  é um conjunto de pontos de apoio com pesos associados  $(w_k^i, i = 0, \dots, N_s)$  e  $\mathbf{X}_k = (\mathbf{X}_j, j = 0, \dots, k)$  é o conjunto de todos os estados no instante  $k$ . Os pesos são normalizados pela expressão  $\sum_{i=1}^{N_s} w_i = 1$  descritas em [36] a densidade *a posterior* pode-se expressar como a equação 3.17.

$$p(\mathbf{X}_k|\mathbf{Y}_k^1) \approx \sum_{i=1}^{N_s} w_k^i \delta(\mathbf{X}_k - \mathbf{X}_k^i). \quad (3.17)$$

A teoria de amostragem por importância, descrita em [38], assegura que pode-se construir um estimador, onde cada amostra  $X_i$  e pesos são apresentados de acordo com as equações 3.18 e 3.19.

$$\mathbf{X}_j^i \propto q(\mathbf{X}_k^i|\mathbf{X}_{k-1}^i, \mathbf{Y}_k^i) \quad (3.18)$$

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{Y}_k|\mathbf{X}_k^i)p(\mathbf{X}_k|\mathbf{X}_{k-1}^i)}{q(\mathbf{X}_k^i|\mathbf{X}_{k-1}^i, \mathbf{Y}_k^i)} \quad (3.19)$$

A distribuição  $q(\mathbf{X}_k^i|\mathbf{X}_{k-1}^i, \mathbf{Y}_k^i)$  é conhecida como *densidade de importância*. Uma opção de escolha desta distribuição pode ser intuitivamente definida como  $q(\mathbf{X}_k^i|\mathbf{X}_{k-1}^i, \mathbf{Y}_k^i) = p(\mathbf{X}_k|\mathbf{X}_{k-1}^i)$ , resultando na equação 3.20.

$$w_k^i \propto w_{k-1}^i p(\mathbf{Y}_k|\mathbf{X}_k^i). \quad (3.20)$$

Um problema muito comum neste algoritmo é o fenômeno da degeneração [38], que surge após algumas interações quando as partículas adquirem pesos insignificantes e por consequência disso aumenta-se o custo computacional no esforço de atualização das partículas para representar  $p(\mathbf{X}_k|\mathbf{Y}_k^i)$ . Uma metodologia para medição adequada da degeneração é o tamanho efetivo de amostragem  $N_{eff}$  [36] definida em 3.21.

$$N_{eff} = \frac{N_s}{1 + Var(w_k^*i)}, \quad (3.21)$$

onde  $Var(w_k^*i)$  representa a variância de  $w_k^*i$  que é definido por:

$$w_k^*i = \frac{p(\mathbf{X}_k^i|\mathbf{Y}_k^1)}{q(\mathbf{X}_k^i|\mathbf{X}_{k-1}^i, \mathbf{Y}_k^i)}. \quad (3.22)$$

Devido à complexidade da obtenção destes valores, pode-se utilizar uma estimação aproximada para  $N_{eff}$  a partir da Equação 3.23.

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s} w_k^i} \quad (3.23)$$

Um valor pequeno de  $\hat{N}_{eff}$  significa um grande valor de *degeneração* das partículas. Para evitar este efeito, pode-se escolher um número muito grande de amostras, o que dificulta a etapa computacional.

### 3.3.2 Reamostragem

O processo de Reamostragem tem como propósito eliminar as partículas consideradas fracas (ou com pouco peso) e manter aquelas com maior importância (ou com grande peso). Para realizar este processo, utiliza-se a substituição das partículas fracas por aquelas com maior peso, por meio de um processo probabilístico que permita obter um outro conjunto de amostras que melhor representem a função discretizada de  $p(\mathbf{X}_k | \mathbf{Y}_k)$ , conforme apresentado na Equação 3.24.

$$p(\mathbf{X}_k | \mathbf{Y}_k^1) \approx \sum_{i=1}^{N_s} w_k^i \delta(\mathbf{X}_k - \mathbf{X}_k^i) \quad (3.24)$$

O resultado da reamostragem é um novo conjunto de partículas com peso uniforme  $1/N_s$ . Existem diversas alternativas para implementação de algoritmos de reamostragem em filtros de partículas, como a amostragem estratificada ou a amostragem residual [39]. O principal fator de análise desses algoritmos é o grau de dificuldade computacional na sua implementação.

A seguir, apresenta-se o método de reamostragem proposto em[40], onde U representa uma distribuição uniforme. Este algoritmo apresenta uma baixa complexidade computacional e alta eficiência.

---

**Algoritmo 1:** Algoritmo de Reamostragem

---

**Entrada:**  $[\{\mathbf{X}_k^{j*}, w_k^j, i^j\}_{j=1}^{N_s}]$ **Saída:**  $[\{\mathbf{X}_k^i, w_k^i\}_{i=1}^{N_s}]$ **início**Inicialização da Função Densidade Probabilidade (FDP)  $c_1 = 0$ **para**  $i = 2 : N_s$  **faça**| obtenção da FDP:  $c_i = c_{i-1} + w_k^i$ **fim para**Escolha de um ponto de início:  $u_1 \sim U[0, N_s^{-1}]$ **para**  $j = 1 : N_s$  **faça**| Percorrer a FDP:  $u_j = u_1 + N_s^{-1}(j - 1)$ **enquanto**  $u_j > c_i$  **faça**|  $i = i + 1$ **fim enqto**| Atribua as amostras  $\mathbf{X}_k^{j*} = \mathbf{X}_k^i$ | Atribua os pesos  $w_k^j = N_s^{-1}$ | Atribua os parentes  $i^j = i$ **fim para****fim**

---

### 3.3.3 Algoritmo de Amostragem e Reamostragem por Importância - SIR

Com o passo de reamostragem, geram-se problemas relativos à diversidade das partículas, pois as partículas de maior peso podem ser escolhidas diversas vezes. Outro problema encontrado com a reduzida diversidade das partículas, é a dificuldade de obtenção de uma estimativa suavizada para a trajetória das partículas. Para evitar estes problemas, o Filtro de Partículas emprega a amostragem por importância e a etapa de reamostragem é ativada dependendo do limite  $N_T$  atribuído para  $N_{eff}$  para evitar o processo de perda da diversidade. Com a reamostragem melhora-se o problema de *degeneração*, concentrando o algoritmo nas partículas que tem os melhores pesos, e que resultam em uma melhor estimação. O algoritmo 2 representa de forma condensada os passos para conseguir um filtro de partículas eficiente.

Em [36], são apresentadas variações do Filtro de Partículas, sempre na tentativa de melhorar o desempenho em relação ao efeito da *degeneração* e a perda de diversidade das partículas.

---

**Algoritmo 2:** Algoritmo de Amostragem e Reamostragem por Importância

---

**Entrada:**  $[\{\mathbf{X}_{k-1}^i, w_{k-1}^i\}_{j=1}^{N_s}, \mathbf{Y}_k]$

**Saída:**  $[\{\mathbf{X}_k^i, w_k^i\}_{i=1}^{N_s}]$

**início**

**para**  $i = 1 : N_s$  **faça**

    | Gera-se  $\mathbf{X}_k^i \sim p(\mathbf{X}_k | \mathbf{X}_{k-1}^i)$

    | Cálculo do  $w_k^i = p(\mathbf{Y}_k | \mathbf{X}_k^i)$

**fim para**

Cálculo dos pesos totais  $t = \sum_{i=1}^{N_s} w_k^i$

**para**  $j = 1 : N_s$  **faça**

    | Normalizar  $w_k^i = w_k^i t^{-1}$

**fim para**

Cálculo do  $\hat{N}_{eff}$

**se**  $\hat{N}_{eff} < N_T$  **então**

    | Reamostragem empregando o algoritmo 1.

**fim se**

**fim**

---

### 3.4 FILTRO DE PARTÍCULAS BASEADO NO MÉTODO DE MONTE CARLO VIA CADEIAS DE MARKOV

O filtro de partículas baseado no método de Monte Carlo via cadeias de Markov (MCMC) difere, basicamente, na forma como são geradas as partículas. No caso do Filtro de Partículas SIR, as partículas são geradas de forma aleatória e são independentes e o grau de êxito depende da densidade de partículas geradas. No caso do Filtro de Partículas MCMC, a geração das partículas se faz por um processo iterativo, conseguindo assim partículas dependentes do estado de transição. O método empregado para gerar Filtros de Partículas inspirados em cadeias de Markov utilizam o algoritmo de Metropolis-Hassting [41] descrito na Seção 3.4.1 a seguir.

#### 3.4.1 Algoritmo de Metropolis-Hassting

O algoritmo Metropolis-Hassting utiliza uma distribuição  $q(X|X^*)$ , onde  $X^*$ , é um candidato que poderá ser aceito ou rejeitado de acordo à densidade  $\alpha(X|X^*)$  que pode ser descrita como:

$$\alpha(\mathbf{X}|\mathbf{X}^*) = \begin{cases} r(\mathbf{X}|\mathbf{X}^*) & \text{se } r(\mathbf{X}|\mathbf{X}^*) \leq 1 \\ 1 & \text{se } r(\mathbf{X}|\mathbf{X}^*) > 1 \end{cases} \quad (3.25)$$

onde,

$$r(\mathbf{X}|\mathbf{X}^*) = \frac{\pi(\mathbf{X}^*)q(\mathbf{X}|\mathbf{X}^*)}{\pi(\mathbf{X})q(\mathbf{X}^*|\mathbf{X})}. \quad (3.26)$$

Se o candidato for aceito, este ocupará uma nova posição, se for rejeitado manterá sua posição atual no espaço de estados. Considerando a posição independente do estado da cadeia de Markov, a amostragem independente para a distribuição proposta  $q(X^*|X^i) = q(X^*)$  pode ser definida pela Equação 3.27.

$$\alpha = \min \left( 1, \frac{Z(\mathbf{X}^*)}{Z(\mathbf{X}^i)} \right) \text{ onde, } Z(x) = \frac{\pi(\mathbf{X})}{q(\mathbf{X})}. \quad (3.27)$$

Para o algoritmo de Metrópolis-Hassting com uma distribuição proposta simétrica  $q(\mathbf{X}^*|\mathbf{X}^i) = q(\mathbf{X}^i|\mathbf{X}^*)$ , a probabilidade de aprovação fica definida como:

$$\alpha = \min \left( 1, \frac{\pi(\mathbf{X}^*)}{\pi(\mathbf{X}^i)} \right) \quad (3.28)$$

### 3.4.2 Estimação

Logo após a geração das partículas, passa-se ao processo da estimação da distribuição probabilidade *a posteriori*  $\pi(\mathbf{X})$  descrita por:

$$\pi(\mathbf{X}_k) = p(\mathbf{Y}_k|\mathbf{X}_k) \sum_i^n p(\mathbf{X}_k|\mathbf{X}_{k-1}), \quad (3.29)$$

onde  $p(\mathbf{Y}_k|\mathbf{X}_k)$  é a função de distribuição do modelo de observação e  $p(\mathbf{X}_k|\mathbf{X}_{k-1})$  é função de distribuição do modelo de movimento *a priori*. O valor do estado  $\mathbf{X}_k$ , pode ser estimado como uma aproximação de *Máximo a posteriori* (MAP), como:

$$\hat{\mathbf{X}} = \mathbf{X}_k^{i_{max}} \quad (3.30)$$

$$i_{max} = \underset{i}{\operatorname{argmax}} \{ \pi(\mathbf{X}_k^i) \} \quad (3.31)$$

## 3.5 DISTÂNCIA BHATTACHARYYA

De acordo com [42], considerando duas populações multinomiais com  $N$  categorias de classes com probabilidades associadas  $p_1, p_2, \dots, p_N$  e  $p'_1, p'_2, \dots, p'_N$  respectivamente, seja  $\sum_i^N p_i = 1$  e  $\sum_i^N p'_i = 1$ . Nota-se que  $(\sqrt{p_1}, \dots, \sqrt{p_N})$  e  $(\sqrt{p'_1}, \dots, \sqrt{p'_N})$  podem ser consideradas como os cossenos diretores de dois vetores no espaço N-dimensional. Como medida

da divergência entre duas populações emprega-se o quadrado do ângulo entre dois vetores de posição. Seja  $\theta$  o ângulo entre os vetores, então de acordo com [43]), têm-se:

$$\cos(\theta) = \sum_i^N \sqrt{p_i p'_i}. \quad (3.32)$$

A medida Bhattacharyya pode ser usada para comparar a similaridade de dois histogramas. Sejam  $R_i$   $S_i$  as frequências normalizadas no *bin*  $i$  de dois histogramas, então, podemos assumir que  $R_i$  e  $S_i$  são distribuições de uma variável aleatória, então a medida Bhattacharyya será:

$$MB = \sum_i \sqrt{R_i} \sqrt{S_i} \quad (3.33)$$

Para dois histogramas idênticos  $MB = 1$  o que indica uma correspondência perfeita. Se  $MB = 0$  indica uma divergência total, não existindo nenhuma similaridade entre os histogramas avaliados.

Após finalizar esta apresentação dos fundamentos probabilísticos aplicados neste trabalho, o Capítulo 4 a seguir descreve a metodologia proposta de elaboração do rastreamento visual utilizando Filtro de Partículas.

## 4 METODOLOGIA PROPOSTA

No decorrer deste capítulo, são apresentadas as etapas componentes do sistema de rastreamento visual, destacando-se a escolha de cada uma das alternativas e melhorias apresentadas. Apresenta-se também a análise da complexidade computacional, com o objetivo de obter uma estimativa da carga computacional em comparação à complexidade computacional da implementação de um Filtro de Partículas em sua forma padrão. A carga computacional da metodologia proposta é essencial na análise da sua viabilidade para utilização em tempo real.

### 4.1 FLUXOGRAMA DO ALGORITMO PROPOSTO

A seguir, apresenta-se o fluxograma dos dados do algoritmo proposto Figura 4.1, cada uma das etapas presentes são desenvolvidas ao longo das secciones da metodologia.

### 4.2 MODELO E DISTRIBUIÇÃO DE CORES

Esta metodologia proposta foi elaborada para realizar o rastreamento visual a partir de informações baseadas no modelo de cor HSV cujas distribuições de cor melhoram a robustez contra situações de não-rigidez, rotação e oclusão parcial do objeto a ser rastreado no domínio da image.

#### 4.2.1 Modelo HSV

O modelo HSV é formado por três componentes : *H(hue)* matiz, *S (saturation)* saturação e *V (value)* valor ou brilho. A *matiz*, representa uma cor do espectro, e possui valores entre 0 e 360 graus, dentro de um círculo cromático (Figura 4.2). O valor de 0 graus corresponde à cor vermelha, enquanto 120 graus corresponde ao azul e 240 graus corresponde ao verde. A *saturação* varia entre 0 e 100% e representa o grau de pureza da cor: quanto maior o valor da saturação, maior a pureza; quando este valor é muito pequeno, a cor tende à escala cinza. O canal *valor* pode variar entre 0 e 100%, representando a intensidade de brilho da cor. Quanto maior o *valor*, mais claras tornam-se as cores, o valor zero representa a cor preta.

A partir da escolha do modelo de cor, esta metodologia propõe incluir apenas os planos H e S, já que esses planos possuem maior informação relativa ao objeto a ser rastreado, permitindo segmentar o objeto das regiões ao seu entorno. Para alcançar maior estabilidade nas amostras, utilizou-se um histograma combinado dos planos H e S com 10 *bins* para cada

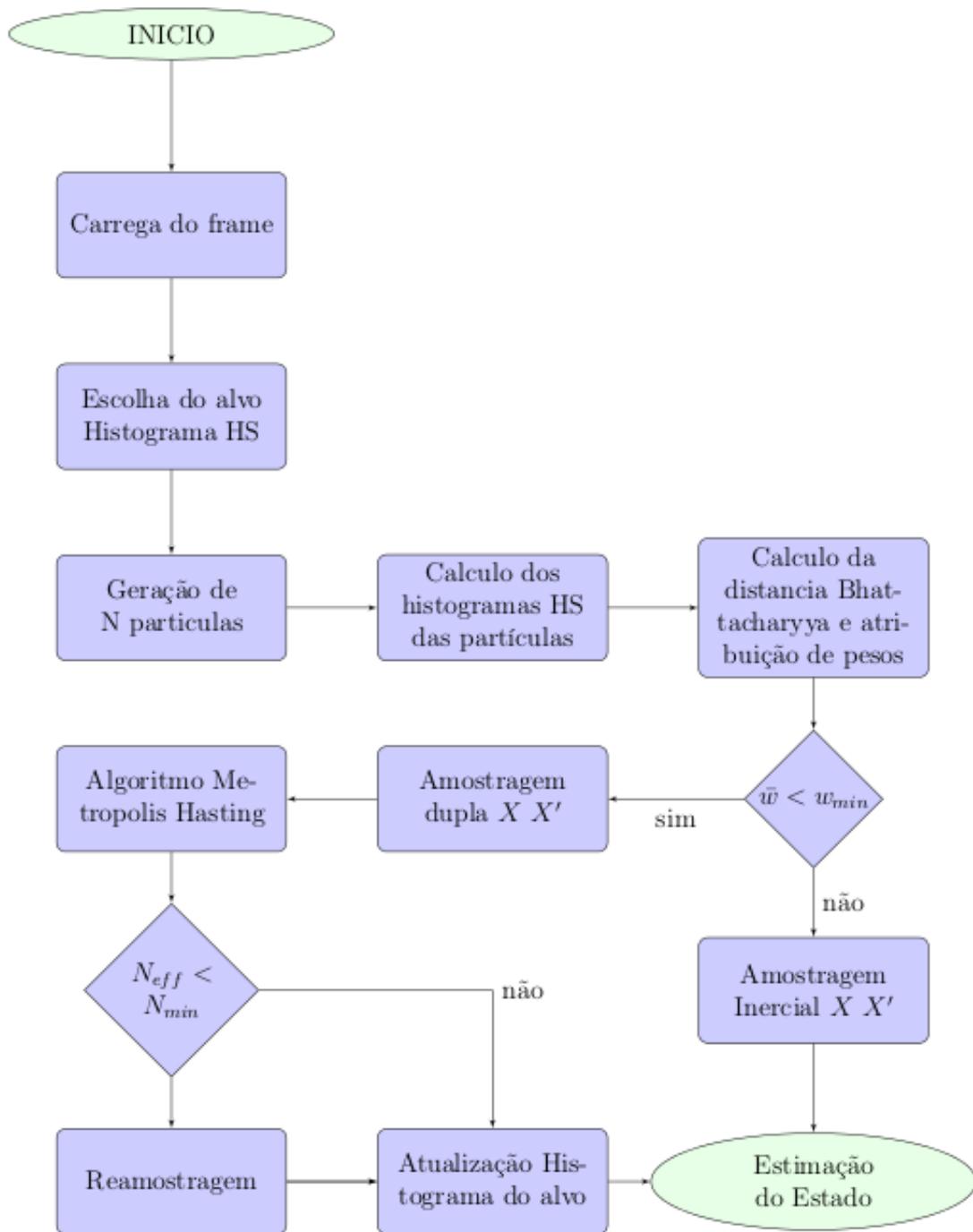


Figura 4.1: Fluxograma do algoritmo proposto.

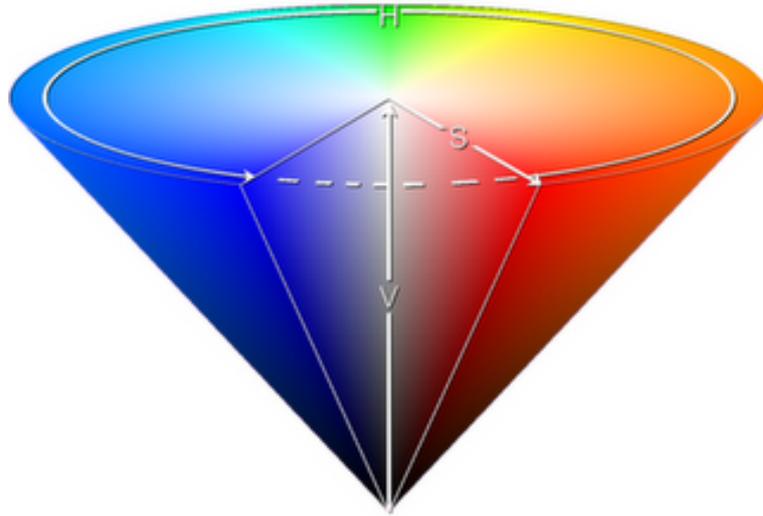


Figura 4.2: Modelo de Cores HSV (retirado de [5]).

um. A divisão da distribuição de cor em um histograma é uma maneira de discretizá-la. O histograma necessita ser normalizado para facilitar a sua comparação e sua atualização ponderada. A figura 4.3 mostra um exemplo de histograma normalizado dos planos H-S de um objeto a ser rastreado.

### 4.3 ATRIBUIÇÃO DE PESOS

Para cada uma das amostras geradas é realizado o cálculo do histograma H-S da área representada  $H^i$ , e em seguida é realizado o cálculo da distância Bhattacharyya com o histograma do objeto rastreado  $H^O$ , a partir da Equação 4.1:

$$d_{H^O-H^i} = \sqrt{1 - MB}, \quad (4.1)$$

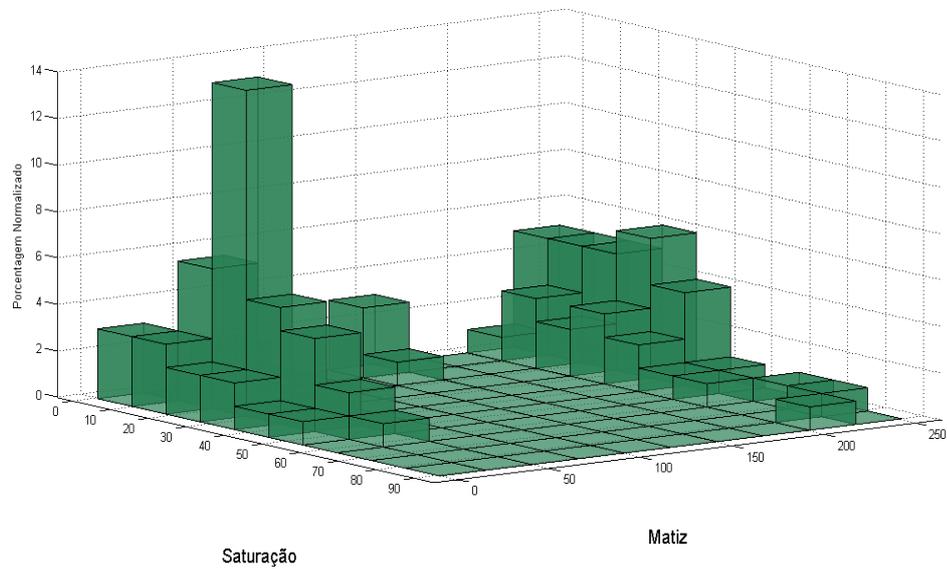
onde MB, é a medida de Bhattacharyya calculada de acordo com a Equação 3.33. Com isso obtém-se uma quantificação da similaridade dos histogramas de cores. Emprega-se esta medida por ser uma relação que não depende do número de *bins* presentes nos histogramas a serem comparados, nem do tamanho de distribuição de cada *bin* de acordo com o análise efetuado em [43]). Este valor é empregado para calcular o peso de cada partícula de acordo com a expressão usada em [44], como:

$$w_i = \exp(-\lambda d_{H^O-H^i}^2), \quad (4.2)$$

em que  $\lambda$  é uma constante, com o valor de 20 (valor sugerido em [44]). A variável



(a)



(b)

Figura 4.3: Histograma H-S: a) Objeto a ser rastreado (envolvido pelo retângulo branco); b) Histograma de matiz e saturação do objeto.

$d_{H^O-H^i}$ , é o valor da distancia Bhattacharyya da amostra  $i$ . Com isto, assegura-se que as amostras possuem maior similaridade com o histograma do alvo terão peso maior, e aquelas com baixa similaridade terão um valor pequeno de peso.

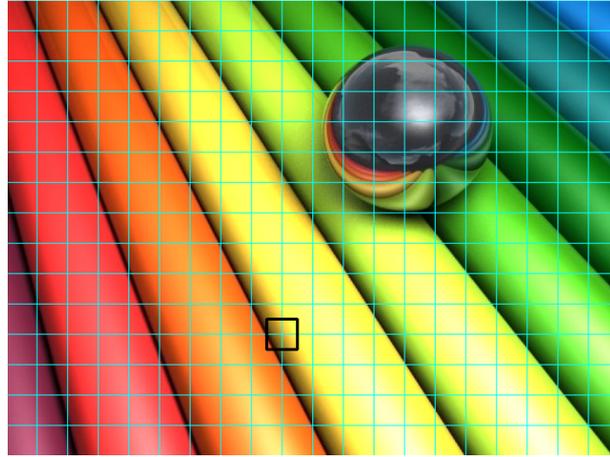
A Figura 4.4 mostra um exemplo em que a imagem de dimensões 700x500 pixels, está dividida em regiões de áreas iguais. Escolhe-se uma amostra (marcada com um quadrado preto), com quatro regiões vizinhas divididas em partes de igual área. É possível observar como a distância Bhattacharyya muda ao longo da imagem, pois um valor muito pequeno significa que o histograma desta área possui muita similaridade como a área padrão escolhida. Nesta abordagem, tem-se uma diferença notável entre as regiões da imagem. A partir da mesma Figura 4.4, observam-se os pesos calculados a partir da Equação 4.2, empregando as divisões anteriores. Neste caso a diferença é notória, obtendo uma melhor segmentação da região de interesse na imagem.

#### 4.4 MODELO DE ATUALIZAÇÃO DO HISTOGRAMA

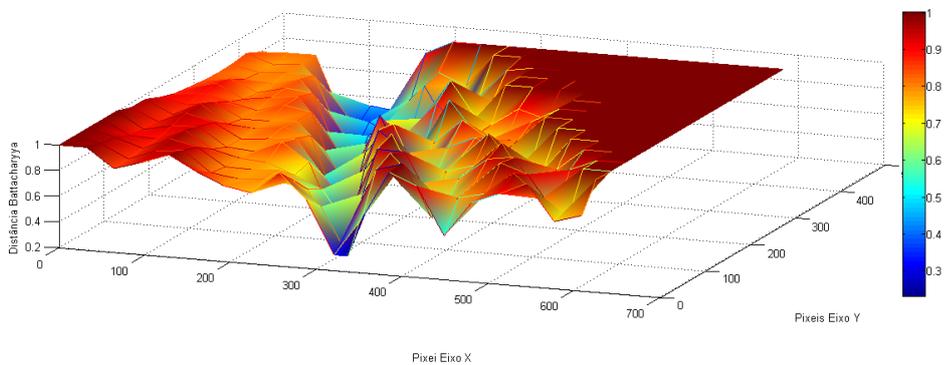
Para fazer o processo de atualização do histograma faz-se o uso do valor médio do peso das partículas, que fornece a informação sobre a relação entre as partículas e o objeto rastreado. Quando este valor está acima do limite  $w_{min}$ , então o histograma do objeto rastreado é atualizado. Este valor também é utilizado para a detecção da oclusão, evitando que o histograma se atualize com valores errados ou adquira valores de cor do entorno (equivalente ao efeito *drift* em técnicas de rastreamento por *template-matching* [25]). Para fazer esta atualização, segue o modelo empregado por Li *et al.* em [45], que propõe uma combinação ponderada das informações dos histogramas do objeto a ser rastreado no instante anterior ( $\mathbf{H}_{k-1}^O$ ), no instante inicial do rastreamento ( $\mathbf{H}_{k=0}^O$ ), da ponderação dos estados das partículas ( $\mathbf{H}_k^{\Sigma w_i \cdot X}$ ) e da partícula com maior peso ( $\mathbf{H}_k^{MP}$ ). Logo, o modelo de atualização pode ser definido na Equação 4.3 a seguir.

$$\mathbf{H}_k^O = (1 - \alpha - \beta - \gamma)\mathbf{H}_{k-1}^O + (\alpha)\mathbf{H}_k^{\Sigma w_i \cdot X} + (\beta)\mathbf{H}_k^{MP} + (\gamma)\mathbf{H}_{k=0}^O, \quad (4.3)$$

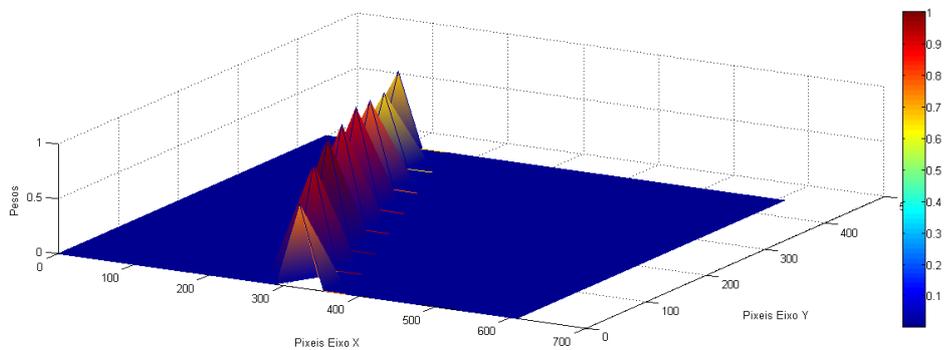
onde os valores de  $\alpha$ ,  $\beta$  e  $\gamma$  são escolhidos de acordo com a velocidade de mudança do alvo. Para o algoritmo proposto,  $\alpha$  é o peso normalizado da estimação das partículas,  $\beta$  é proporcional ao peso normalizado da partícula com maior peso; e por último  $\gamma$  fixa-se em 0.1, levando em consideração que o alvo muda sua aparência durante o rastreamento. Esta abordagem permite melhorias fundamentais na robustez da metodologia em relação às diversas mudanças que podem ocorrer durante a execução do rastreamento.



(a)



(b)



(c)

Figura 4.4: (a)Imagem original dividida em áreas iguais. O quadrado preto representa a área a ser comparada. (b)Valores da distância Bhattacharyya para cada região em comparação com o padrão de comparação. (c)Valores dos pesos das áreas da imagem.

#### 4.5 FILTRO DE PARTÍCULAS POR AMOSTRAGEM DE IMPORTÂNCIA HÍBRIDA

A partir das metodologias apresentadas na Seção 3.3, a proposta desenvolvida nesta dissertação segue o algoritmo baseado no Filtro de Monte Carlo via Cadeias de Markov (Seção 3.4). Esta escolha tem como objetivo aumentar a robustez e implementar melhorias no trata-

mento de oclusões no processo de rastreamento visual. Tanto a melhoria da robustez, quanto a de tratamento de oclusões ocorrem devido a incorporação do comportamento inercial desejado do grupo de amostras do filtro, durante a etapa de reamostragem (Seção 3.3.2). A seguir são apresentados os aspectos gerais do algoritmo e em seguida a descrição mais detalhada das etapas do método proposto.

#### 4.5.1 Estratégia de reamostragem das partículas

Com o objetivo de melhorar a resposta do algoritmo, propõe-se fazer uma divisão do grupo total de partículas  $N_s$ . Cada divisão (grupo) do espaço de partículas é denotada como  $\lambda, \mu, \nu, \dots$  onde  $\lambda + \mu + \nu + \dots = N_s$ . Na descrição de espaço de estados, os vetores  $X^1, X^2, X^3, \dots$  representam os estados associados a cada grupo de partículas. A atualização dos estados depende somente do estado no instante anterior e do fator de correlação  $c_n$  entre os grupos como descrito na Equação 4.4.

$$\begin{aligned} & p(\tilde{\mathbf{X}}_k^A | \mathbf{X}_{k-1}^{A,i}) \\ & p(\tilde{\mathbf{X}}_k^B | \mathbf{X}_{k-1}^{B,i}, \mathbf{c}_1) \\ & p(\tilde{\mathbf{X}}_k^C | \mathbf{X}_{k-1}^{C,i}, \mathbf{c}_2) \\ & \vdots \end{aligned} \quad (4.4)$$

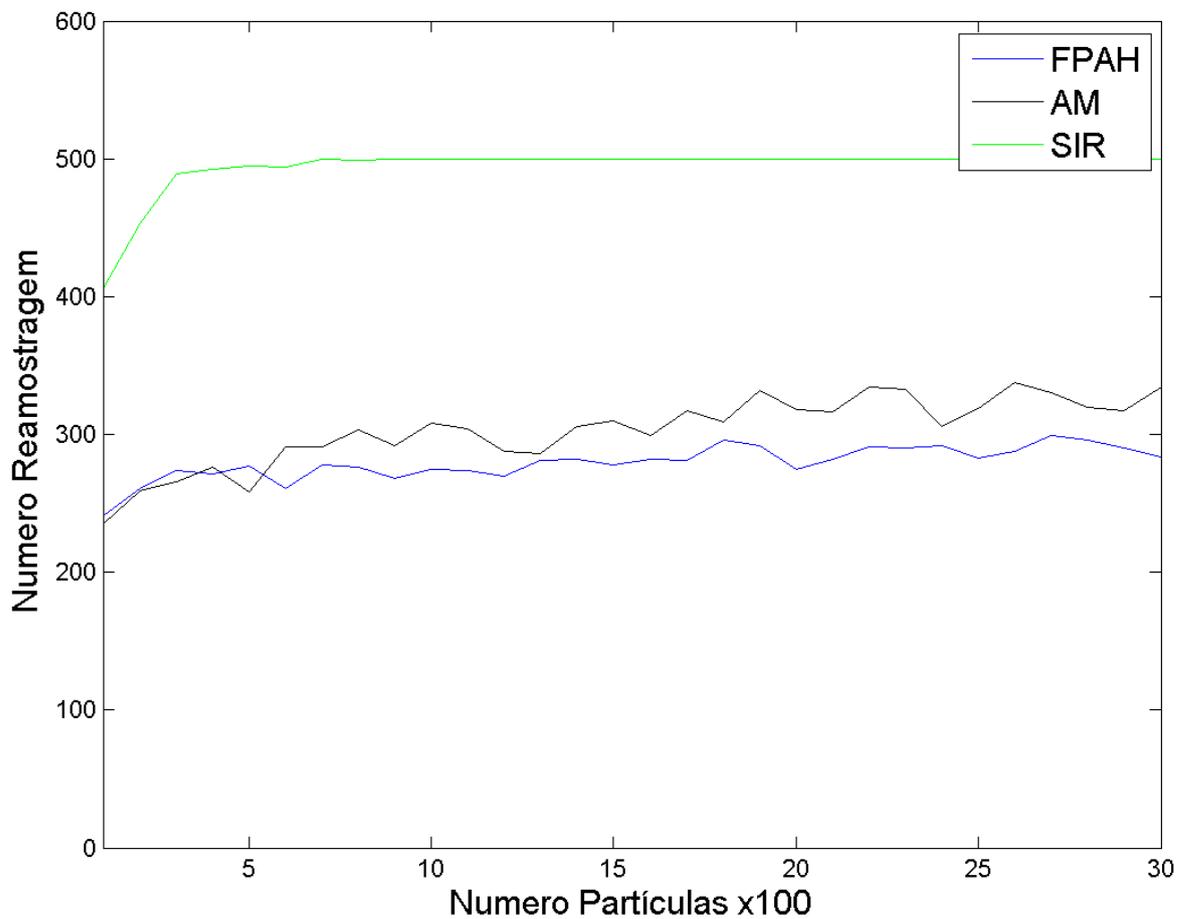
#### 4.5.2 Algoritmo Metropolis-Hasting

Nesta etapa, Cada uma das distribuições de probabilidade, obtidas da predição são submetidas a uma avaliação para definir a convergência das amostras à distribuição de  $X$ . Para isso emprega-se o algoritmo de *Metropolis-Hassting* (Seção3.4.1), fazendo desta forma que as distribuições de probabilidade passem a ser definidas como:

$$\begin{aligned} p(\mathbf{X}_k^A | \mathbf{X}_{k-1}^{A,i}) &= \alpha(\mathbf{X}_k^A | \tilde{\mathbf{X}}_{k-1}^A) p(\tilde{\mathbf{X}}_k^A | \mathbf{X}_{k-1}^{A,i}) \\ p(\mathbf{X}_k^B | \mathbf{X}_{k-1}^{B,i}) &= \alpha(\mathbf{X}_k^B | \tilde{\mathbf{X}}_{k-1}^B) p(\tilde{\mathbf{X}}_k^B | \mathbf{X}_{k-1}^{B,i}, \mathbf{c}_1) \\ p(\mathbf{X}_k^C | \mathbf{X}_{k-1}^{C,i}) &= \alpha(\mathbf{X}_k^C | \tilde{\mathbf{X}}_{k-1}^C) p(\tilde{\mathbf{X}}_k^C | \mathbf{X}_{k-1}^{C,i}, \mathbf{c}_2) \\ & \vdots \\ & \vdots \end{aligned} \quad (4.5)$$

Para melhorar a convergência das amostras, o algoritmo realiza uma classificação das partículas aceitando aquelas que tem um maior valor de probabilidade, representado pelo peso  $w^i$ . Para isso, se faz a comparação dos pesos do estado atual da amostra, estado resultante do amostragem, com o peso da partícula num estado anterior, ou seja, antes do amostragem. Com isso garante-se maior efetividade do processo de amostragem, por quanto são retiradas aquelas partículas fracas. O novo set de partículas tem uma probabilidade alta, o que representa uma melhor estimativa do estado do objeto rastreado.

Para avaliar o emprego do algoritmo de Metrópolis-Hasting, fizeram-se testes para três diferentes tipos de amostragem, com um exemplo simples de rastreamento de um objeto (uma caixa cinza) sobre um fundo azul, a qual possui um movimento senoidal adicionada de um pequeno fator de ruído gaussiano com 500 quadros. São empregados diferente número de partículas empezando desde 100 partículas até 3000, de cada prova se obteve o valor de vezes que é ativado o algoritmo de reamostragem. Se faz a proba para três diferentes tipos de amostragem, a saber: algoritmo probabilístico de rastreamento do filtro de partículas com amostragem por reamostragem de importância (SIR), Filtro de partículas empregando o algoritmo de Metropolis-Hasting (AM) e por ultimo algoritmo proposto (FPAH).



(a)

Figura 4.5: Gráfica do numero de partículas e o numero de vezes que é empregado o processo de reamostragem.

Na figura 4.5, verifica-se que quando o numero de partículas é superior a 700, o algoritmo SIR o ativa o reamostragem durante toda a sequência. A resposta do AM e muito similar ao algoritmo proposto com um número pequeno de partículas, entretanto, o FPAH, mantém um numero de reamostragem menor que o AM para um numero maior de partículas, consequên-

cia de isso, é um tempo de processamento menor, melhorando o desempenho do algoritmo proposto.

### 4.5.3 Distribuição a posteriori

Para definir a distribuição *a posteriori*  $\pi(X)$  de acordo com a regra de *Bayes*, relacionando à distribuição *a priori* com a distribuição obtida das observações de estado, têm-se que:

$$\pi(X) = p(\mathbf{Y}_k | \mathbf{X}_k^i) \left[ \sum_{i=1}^{\lambda} p(\mathbf{X}_k^A | \mathbf{X}_{k-1}^{A,i}) + \sum_{i=\lambda}^{\mu} p(\mathbf{X}_k^B | \mathbf{X}_{k-1}^{B,i}) + \sum_{i=\mu}^{\nu} p(\mathbf{X}_k^C | \mathbf{X}_{k-1}^{C,i}) \dots \right]. \quad (4.6)$$

Para que esta relação seja satisfeita, deve-se cumprir a condição básica de implementação de Filtros de Partículas, descrita na Equação 4.7.

$$\sum_{i=1}^{\lambda} p(\mathbf{X}_k^A | \mathbf{X}_{k-1}^{A,i}) + \sum_{i=\lambda}^{\mu} p(\mathbf{X}_k^B | \mathbf{X}_{k-1}^{B,i}) + \sum_{i=\mu}^{\nu} p(\mathbf{X}_k^C | \mathbf{X}_{k-1}^{C,i}) \dots \leq 1 \quad (4.7)$$

Esta abordagem genérica, descrita na Equação 4.7, permite o desenvolvimento de estratégias diferentes e variadas para o processo de reamostragem, permitindo diversas possibilidades para as situações encontradas no rastreamento visual.

A partir destas premissas, e também com o intuito de validar a metodologia proposta para o algoritmo de rastreamento visual desenvolvido, o grupo de partículas foi dividido em duas partes, denotadas como  $\lambda$  e  $\mu$ , com número igual de partículas. Dependendo do desempenho desejado do algoritmo, podem-se empregar outras divisões do grupo de partículas, incorporando as informações a serem utilizadas no processo de Reamostragem.

O processo de Reamostragem, nesta abordagem, permite que cada grupo represente um comportamento diferente de movimentação, o que aumenta a robustez do rastreamento. O vetor de estados de cada uma das amostras contém os valores de posição em coordenadas  $x$  e  $y$  da cena, as dimensões espaciais do objeto a ser rastreado (comprimento e largura) delimitados pela região de interesse (ex.: retângulo que contém o alvo), além da variação da posição a cada instante de tempo.

O conjunto de partículas  $\lambda$  recebe a informação proveniente do estado  $\mathbf{X}'$  e o atualiza de acordo com a expressão:

$$\tilde{\mathbf{X}}_k^{A,i} = \mathbf{X}_{k-1}^{A,i} + \mathbf{r}, \quad (4.8)$$

onde  $\mathbf{r}$  representa o fator de ruído de transição gaussiano para cada uma das variáveis de estado e  $\tilde{\mathbf{X}}_k$  representa o estado resultante da predição. Já para o conjunto de partículas  $\mu$ , o

processo de atualização é definido pela Equação 4.9:

$$\tilde{\mathbf{X}}_k^B = \mathbf{X}_{k-1}^{B,i} + \mathbf{v} + \mathbf{r}, \quad (4.9)$$

onde a variável  $\mathbf{v}$  representa um fator que descreve os parâmetros do modelo dinâmico do movimento do alvo, denominado componente inercial, e é obtido a partir da expressão:

$$\mathbf{v} = \sum_{i=1}^{\lambda} w_k^i (\mathbf{X}_k^{A,i} - \mathbf{X}_{k-1}^{A,i}). \quad (4.10)$$

Após esta etapa, inicia-se a avaliação do estágio de predição de estados utilizando o algoritmo de Metrópolis-Haasting para garantir a convergência da distribuição de partículas. Esta etapa é descrita para cada conjunto de partículas como:

$$\begin{aligned} p(\mathbf{X}_k^A | \mathbf{X}_{k-1}^{A,i}) &= \alpha(\mathbf{X}_k^A | \tilde{\mathbf{X}}_{k-1}^A) p(\tilde{\mathbf{X}}_k^A | \mathbf{X}_{k-1}^{A,i}), \\ p(\mathbf{X}_k^B | \mathbf{X}_{k-1}^{B,i}) &= \alpha(\mathbf{X}_k^B | \tilde{\mathbf{X}}_{k-1}^B) p(\tilde{\mathbf{X}}_k^B | \mathbf{X}_{k-1}^{B,i}, \mathbf{v}). \end{aligned} \quad (4.11)$$

Para definir  $\pi(\mathbf{X})$ , que representa a distribuição *a posteriori*, tem-se:

$$\pi(X) = p(\mathbf{Y}_k | \mathbf{X}_k^i) \left[ \sum_{i=1}^{\lambda} p(\mathbf{X}_k^A | \mathbf{X}_{k-1}^{A,i}) + \sum_{i=\lambda}^{\mu} p(\mathbf{X}_k^B | \mathbf{X}_{k-1}^{B,i}) \right], \quad (4.12)$$

respeitando a restrição da Equação 4.13:

$$\sum_{i=1}^{\lambda} p(\mathbf{X}_k^A | \mathbf{X}_{k-1}^{A,i}) + \sum_{i=\lambda}^{\mu} p(\mathbf{X}_k^B | \mathbf{X}_{k-1}^{B,i}) \leq 1. \quad (4.13)$$

Se a proporção das partículas  $\lambda$  influenciar a maioria das partículas, o filtro pode perder características inerciais e passa se comportar como se estivesse com uma etapa de predição somente, baseada na estimação da posição anterior. Caso contrário, se a proporção de  $\mu$  sofrer maior influência, então a estimação do movimento das partículas reduz drasticamente sua robustez, mediante à falta da informação inercial. Portanto, a escolha da proporção utilizada na divisão do número de partículas deve ser considerada como um fator altamente influente na estabilidade e convergência do filtro de partículas aqui proposto.

Desta feita, o valor do estado  $\mathbf{X}_k$ , pode ser estimado como uma aproximação do *máximo a posteriori* conforme descrito pelas Equações 4.14 e 4.15.

$$\hat{\mathbf{X}} = \mathbf{X}_k^{i_{max}} \quad (4.14)$$

$$i_{max} = \operatorname{argmax}_i \{ \pi(\mathbf{X}_k^i) \} \quad (4.15)$$

## 4.6 PROCESSO DE ATUALIZAÇÃO DE ESTADOS

Seguindo a abordagem básica de um Filtro de Partículas com reamostragem por importância sequencial, faz-se necessário descrever como ocorrem os dois processos, amostragem e reamostragem, para que se justifique as melhorias aqui apresentadas.

### 4.6.1 Amostragem

O primeiro conjunto denotado pelas partículas com o estado  $\mathbf{X}^A$  é influenciado de acordo com a equação 4.8 com um fator gaussiano  $\mathbf{r}_k$ , em que este representa uma matriz de covariâncias para os valores da posição das amostras e da área que envolve o objeto a ser rastreado. A partir de uma distribuição gaussiana com base na restrição de suavidade de movimento, ou seja, assume-se que o movimento do objeto entre um quadro e outro da imagem é relativamente pequeno, as amostras precisam ser geradas a partir da última posição conhecida para tentar estimar sua nova posição no quadro seguinte. Igualmente ocorre com as informações referentes às mudanças de escala do objeto, em que adota-se a aproximação de que o tamanho do objeto não sofre grandes variações entre os quadros da imagem. Desta forma as novas amostras assumem a última estimacão e depois são geradas novas amostras com as novas posições próximas as posições antigas e com escalas quase iguais.

O Segundo grupo denotado por  $\mathbf{X}^B$  segue a suposição de que a velocidade do objeto é quase constante entre quadros (restrição de suavização de velocidade), e ainda, esta velocidade é extraída pelo comportamento das amostras  $\mathbf{X}^A$ . Com esta abordagem deseja-se aumentar a inércia intrínseca do conjunto de partículas em relação a velocidade estimada do objeto. Neste conjunto também é mantida a aproximação da escala do objeto adicionado a um pequeno fator de escala espacial, que fornece a informação sobre a variação da escala espacial do objeto. A nova posição das partículas  $\mathbf{X}^B$  é atualizada adicionando à posição anterior a velocidade  $\mathbf{v}$  obtida pela Equação 4.10 e ponderada por  $\mathbf{X}^A$ , adicionado pelo fator proveniente de uma distribuição gaussiana com média zero e desvio padrão pré-definido.

Na etapa de amostragem inicia-se o algoritmo de *Metropolis-Haasting* atuando em todo o conjunto de amostras, fazendo uma comparação dos pesos anteriores e dos pesos depois da atualização de estados. Com esta estratégia se garante que a nova posição  $x$  e  $y$  e escala das partículas convergem a estimacão. Portanto, tem-se uma amostra candidata  $\mathbf{X}^*$  com a posição e escala da região de interesse que contém o objeto, descrito pela Equação 4.16.

$$\alpha = \min \left( 1, \frac{w^*}{w^i} \right). \quad (4.16)$$

Em que  $w_i^*$  é o peso associado à partícula candidata e  $w_i$ , o peso da partícula no instante anterior. Se a nova amostra gerada tiver um maior peso, ou seja, uma maior probabilidade de representação do alvo, então esta é aceita no processo de amostragem. Caso contrário, então

o estado estimado no instante de tempo anterior representará a melhor opção de amostragem.

#### 4.6.2 Reamostragem

Esta etapa é empregada quando existe uma degeneração elevada dos pesos das partículas. Desta forma executa-se um procedimento que procura manter as partículas com maior peso e rejeitar aquelas com um peso muito pequeno. Esta abordagem permite-se concentrar as amostras em torno da melhor estimativa de posição espacial e escala do objeto a ser rastreado. Neste trabalho, utiliza-se o algoritmo de Reamostragem descrito em no Capítulo 3.

A título de exemplificar o comportamento da atualização de estados e pesos de partículas propostas, é apresentado um exemplo simples de rastreamento de um objeto (uma caixa cinza), a qual possui um movimento senoidal adicionada de um pequeno fator de ruído gaussiano. Neste caso são apresentadas somente o comportamento das posições das partículas durante a evolução do rastreamento.

Na Figura 4.6 (a) as amostras no momento inicial tem a posição central da imagem. A primeira sequência possui uma amostragem gaussiana para as coordenada  $x$  e  $y$  da imagem. É possível observar as partículas expandindo em torno do ponto central, até que o alvo fica muito próximo delas. Então o algoritmo de Reamostragem consegue reuni-las para as proximidades do objeto a ser rastreado.

A segunda sequência (Figura 4.6 (b)) também possui a mesma amostragem utilizando o algoritmo de rejeição e aceitação, que faz com que as partículas se movimentem mais rapidamente, indo ao encontro do alvo possibilitando uma melhoria no desempenho do rastreamento.

Por último, na sequência da Figura 4.6 (c), é obtida mostrando a utilização da metodologia de Reamostragem proposta, em que consiste em alimentar um grupo de partículas com as informações de inércia das partículas e com as informações da melhor partículas estimadas. Na Figura 4.6 (c) o grupo com as informações de  $X^1$  (cor azul) tem amostragem com a distribuição da melhor partícula, enquanto que o grupo de partículas  $X^2$  (cor vermelha) tem velocidade igual à resultante da ponderação de velocidade das partículas do grupo azul. Com esta estratégia, facilmente pode-se observar que as partículas vermelhas são rapidamente convergem ao objeto, conseguindo um desempenho melhor que os dois algoritmos de amostragem anteriores.

### 4.7 DETECÇÃO E TRATAMENTO DE OCLUSÕES

A partir do exposto nas Seções anteriores, é possível definir um valor de pesos  $w_{min}$  que pode ser utilizado para determinar quando uma oclusão está ocorrendo. Para tal é obtido o valor médio dos pesos, e se eles são menores que  $w_{min}$ , logo existe a oclusão. Este com-

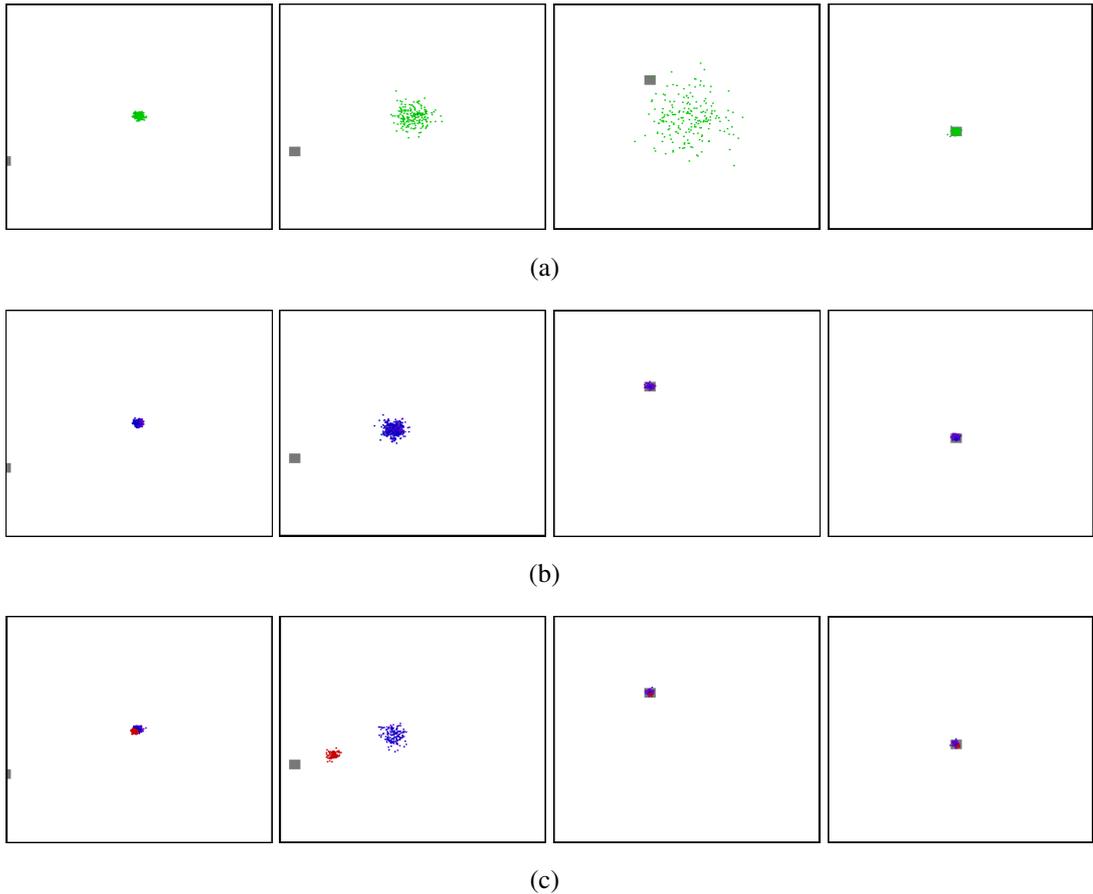


Figura 4.6: Exemplo conceitual de rastreamento de um objeto com movimento senoidal com diferentes amostragens. a) Amostragem padrão. b) Amostragem com Algoritmo Metropolis-Hasting. C) Amostragem sequencial por importância Híbrida com Algoritmo Metropolis-Hasting.

portamento ocorre devido as partículas adquirem valores de pesos baixos devido a diferença entre o histograma do alvo e o histograma obtido na área que está ocultando ao objeto.

Neste momento, as partículas provenientes de  $X^B$  seguem atualizando seu estado de acordo com a última velocidade obtida no instante de tempo anterior à oclusão, para que desta forma mantenham o comportamento inercial, estimando a possível trajetória do alvo. As amostras de  $X^A$  permanecem no entorno da última estimativa da posição do alvo com o propósito de encontrar o objeto caso exista outro objeto em movimento, e também, caso este objeto esteja obstruindo o alvo e este permaneça fixo. Durante a oclusão o histograma do alvo permanece sem nenhuma variação até que a detecção do objeto volte a ocorrer.

#### 4.8 PSEUDOCÓDIGO DA METODOLOGIA PROPOSTA

Reunindo todos os aspectos e características relatados nas Seções anteriores, foi possível a obtenção de um algoritmo rápido e eficiente que permite realizar o rastreamento de objetos

em sequência de imagens. A estrutura fundamental deste algoritmo é inspirada em um Filtro de Partículas realizando Amostragem seqüencial por importância, em conjunto com um processo de Reamostragem híbrida. As etapas da metodologia implementada proposta são descritas no Algoritmo 3, em que a entrada do algoritmo é uma região de interesse (ROI) que envolva o objeto/alvo a ser rastreado no domínio da imagem. E a saída é a estimativa da posição do objeto/alvo no quadro seguinte da sequência de imagens.

---

**Algoritmo 3:** Algoritmo Proposto

---

**Entrada:**  $[\{\mathbf{X}_{k-1}^i, w_{k-1}^i\}_{j=1}^{N_s}, \mathbf{Y}_k]$

**Saída:**  $[\{\mathbf{X}_k^i, w_k^i\}_{i=1}^{N_s}]$

**início**

**para**  $i = 1 : \frac{N_s}{2}$  **faça**

| Gera-se  $\mathbf{X}_k^{A,i} \sim p(\mathbf{X}_k^A | \mathbf{X}_{k-1}^{A,i})$  Eq. 4.8;

| Cálculo do  $w_k^{*,i} = p(\mathbf{Y}_k | \mathbf{X}_k^i)$  Eq. 4.2;

**fim para**

**para**  $i = \frac{N_s}{2} : N_s$  **faça**

| Gera-se  $\mathbf{X}_k^{B,i} \sim p(\mathbf{X}_k^B | \mathbf{X}_{k-1}^{B,i}, \mathbf{v})$  Eq. 4.9;

| Cálculo do  $w_k^{*,i} = p(\mathbf{Y}_k | \mathbf{X}_k^i)$  Eq. 4.2;

**fim para**

**para**  $i = 1 : N_s$  **faça**

| **se**  $1 < \frac{w^{*,i}}{w^i}$  **então**

| |  $\mathbf{X} = \mathbf{X}^*$ ;

| |  $w^i = w^{*,i}$ ;

| **senão**

| |  $\mathbf{X} = \mathbf{X}_{k-1}$ ;

| |  $w^i = w_{k-1}^i$ ;

| **fim se**

**fim para**

Cálculo da velocidade  $\mathbf{v}$  Eq.4.10;

Cálculo dos pesos totais  $t = \sum_{i=1}^{N_s} w_k^i$ ;

**para**  $j = 1 : N_s$  **faça**

| Normalizar  $w_k^i = w_k^i t^{-1}$ ;

**fim para**

Cálculo do  $\hat{N}_{eff}$  Eq.3.23;

**se**  $\hat{N}_{eff} < N_{lim}$  **então**

| Reamostragem empregando o Algoritmo 1;

**fim se**

Atualização do Histograma do Alvo Eq. 4.3;

Estimação do estado Eq.4.14;

**fim**

---

Na parte inicial do algoritmo, a entrada é o conjunto de amostras com pesos associados. A saída é outro conjunto de amostras representando a função de densidade probabilidade da posição e o tamanho do objeto a ser rastreado. O primeiro passo faz-se a atualização dos estados das amostras. Cada um dos conjunto de amostras se atualizam de forma independente: O primeiro com uma distribuição gaussiana e o segundo com os valores inerciais obtidos no instante anterior.

O segundo estágio do algoritmo é o passo de aceitação ou rejeição dos pesos das partículas, que é executado pela comparação dos pesos das amostras iniciais e as amostras com os estados atualizados, em que a melhor estimativa permanece no conjunto de amostras. Definida a melhor estimativa, calcula-se os valores inerciais que serão utilizados no passo seguinte para realizar uma nova atualização do estados das amostras. Em seguida normalizam-se os pesos e calcula-se o valor de  $N_{eff}$  para determinar o grau de proximidade das amostras com o objeto. Dependendo do valor de  $N_{eff}$ , utiliza-se o algoritmo de Reamostragem. O passo seguinte é a atualização do histograma de acordo com a Equação 4.3, seguindo o modelo de atualização ponderado. Por ultimo realiza-se a estimação da posição e as dimensões do alvo para o próximo quadro da imagem.

#### 4.9 ANÁLISE DA COMPLEXIDADE COMPUTACIONAL DO ALGORITMO PROPOSTO

Para fazer a análise de complexidade computacional, a partir de cada uma das linhas de instruções atribui-se uma variável que represente o tempo de execução daquela instrução. Se a instrução esta dentro de um ciclo de repetição (i. e. *while, for, ...*), a ordem de complexidade aumenta proporcionalmente em relação aos ciclos concatenados. A variável  $n$  representa quantidade de partículas de entrada, neste caso o número de amostras empregadas. Esta análise ocorre conforme a metodologia empregada em [46].

Instrução	Custo C.	Repetições
1. <b>Para</b> $i = 1 : \frac{N_s}{2}$	$C_1$	$\frac{n}{2}$
2. $\mathbf{X}_k^{A,i} \sim p(\mathbf{X}_k^A   \mathbf{X}_{k-1}^{A,i})$	$C_2$	$\frac{n-1}{2}$
3. $w^{*,i} = P(\mathbf{Y}_k   \mathbf{X}_k^i)$	$C_3$	$\frac{n-1}{2}$
4. <b>Para</b> $i = \frac{N_s}{2} : N_s$	$C_4$	$\frac{n}{2}$
5. $\mathbf{X}_k^{B,i} \sim p(\mathbf{X}_k^B   X_{k-1}^{B,i}, \mathbf{v})$	$C_5$	$\frac{n-1}{2}$
6. $w^{*,i} = P(\mathbf{Y}_k   \mathbf{X}_k^i)$	$C_6$	$\frac{n-1}{2}$
7. <b>Para</b> $i = 1 : N_s$	$C_7$	$n$
8. <b>Se</b> $1 < \frac{w^{*,i}}{w_i}$	$C_8$	$n$
9. $\mathbf{X} = \mathbf{X}^*$	$C_9$	$n$
10. $w^i = w^{*,i}$	$C_{10}$	$n$
11. <b>senão</b>	$C_{11}$	$n$

12.	$\mathbf{X} = \mathbf{X}_{k-1}$	$C_{12}$	$n$
13.	$w^i = w_{k-1}^{*,i}$	$C_{13}$	$n$
14.	<b>Para</b> $i = 1 : \frac{N_s}{2}$	$C_{14}$	$\frac{n}{2}$
15.	$\vec{v} = \sum_i^{N_s} w^i \Delta X$	$C_{15}$	$\frac{n-1}{2}$
16.	<b>Para</b> $i = 1 : N_s$	$C_{16}$	$n$
17.	$t = \sum_i^{N_s} w_k^i$	$C_{17}$	$n - 1$
18.	$N = \sum_i^{N_s} (w_k^i)^2$	$C_{18}$	$n - 1$
19.	$N_{eff} = \frac{1}{N}$	$C_{19}$	1
20.	<b>Se</b> $N_{eff} < N_{lim}$	$C_{20}$	1
21.	<b>Re-amostragem</b>	$C_{21}$	$n$
22.	<b>Atualização Histograma</b>	$C_{22}$	1
23.	<b>Estimação</b>	$C_{23}$	1

Quando se tem ciclos simples, ou seja sem repetições, o custo computacional para uma entrada de tamanho  $n$  é proporcional a este valor linearmente. Para a etapa condicional na linha 8, considera-se o pior dos casos para a avaliação, portanto, atribui-se o custo de  $n$  para cada uma das condições a serem satisfeitas. Na linha 21 executa-se a condição  $N_{eff} < N_{lim}$ , em que no pior dos casos todas as vezes que o algoritmo de *Reamostragem* é necessário durante todo o processo. O valor da complexidade computacional do processo de *Reamostragem* é  $O(n)$  (detalhes da obtenção desta complexidade são descritos anexo A.3). Finalizando a análise, obtém-se o valor da complexidade computacional  $C_c$  como a soma de todas as instruções, sintetizados pela Equação 4.17.

$$C_c = An + B \quad (4.17)$$

Esta é uma forma simplificada resultante. A variável  $A$  representa todas as variáveis do custo computacional associado com o desempenho da maquina na qual roda o algoritmo,  $B$  representa as variáveis das instruções que não dependem do numero de partículas para serem executados. A ordem máxima obtida para  $n$  é 1(um), pois o fator de complexidade é completamente do número de partículas de entrada, tendo um acréscimo considerado de carga computacional mediante a escolha do número de partículas a serem utilizadas pelo Filtro. De acordo com estas premissas, faz-se notado em uma complexidade computacional para o algoritmo definido por  $O(n)$ , sendo  $O$  um fator independente do número das amostras. Comparando com o valor obtido da análise do Filtro de partículas padrão com amostragem sequencial por importância (vide análise no anexo A.2), o algoritmo aqui proposto tem a mesma complexidade computacional que o filtro de partículas padrão, mesmo com as abor-

dagens realizadas no processo de Amostragem e Reamostragem propostos.

## 5 RESULTADOS DA IMPLEMENTAÇÃO

Neste capítulo são apresentados os resultados obtidos com a metodologia proposta no capítulo anterior. O desempenho do algoritmo proposto é demonstrado em sequências que representam situações comuns, que exigem maior robustez no rastreamento visual de objetos.

Para fins de comparação são apresentados em conjunto os resultados utilizando dois algoritmos de rastreamento visual: Algoritmo *WM* (Window-Matching) disponibilizado na distribuição oficial do OpenCV [47]; e a versão do Filtro de Partícula com amostragem por importância proposto em [36].

Foram utilizados para os diferentes testes, os vídeos disponíveis no *Bonn Benchmark on Tracking* - BoBot[48]. Este repositório disponibiliza diversas sequências classificadas com diferentes tipos de dificuldade para rastreamento de objetos (ex. pessoas, canecas, etc) em diversas situações de iluminação, variação de escala espacial e oclusões. As sequências possuem as dimensões espaciais de  $320 \times 250$  pixels e a taxa de  $25 \text{ fps}$ . Além da variedade de situações disponíveis para a verificação das metodologias de rastreamento visual deste *Benchmark*, o fator que mais influenciou em sua escolha foi a disponibilidade de *Ground Truth* exclusivamente desenvolvido para aplicações em rastreamento. Este *Ground Truth* disponibiliza para cada uma das sequências os dados de um retângulo (largura e altura) que encerra o contorno do objeto alvo, o que facilita a verificação da metodologia proposta.

Com o intuito de testar e mensurar a eficácia e robustez da metodologia proposta, empregou-se como critério que os algoritmos baseados em Filtro de Partículas, seja no SIR (*Sampling Importance resampling* amostragem e reamostragem por importância) ou utilizando reamostragem híbrida (definido pela sigla FPAH - Filtro de Partículas com Amostragem Híbrida) utilizam a quantidade fixa de 200 partículas. No caso do Filtro FPAH estabeleceu-se a distribuição de metade das partículas seriam atualizadas no processo de reamostragem híbrida pela estimativa da melhor partícula. A metade remanescente seria atualizada pela informação inercial do movimento.

A implementação foi desenvolvida em linguagem C++ utilizando a biblioteca *opensource* OpenCV nas etapas de visualização das sequências de imagens e cálculo de histogramas [49]. A arquitetura computacional utilizada para os testes da aplicação foi um computador com processador Intel Core i5, com 4 Gb de memória RAM, sistema operacional Linux, distribuição Ubuntu 11.04.

Os testes são descritos nas seções a seguir abordando as seguintes situações, a saber: *Rastreamento simples sem oclusão; com variação de escala espacial; variação de trajetória e velocidade do objeto; variação de background; variação de iluminação; oclusões parciais e totais; e múltiplos objetos*. Após realizada a etapa de testes em diversas situações, são apresentados resultados do rastreamento para vários alvos e também comparativos entre técnicas

de rastreamento visual.

## 5.1 RASTREAMENTO SIMPLES SEM OCLUSÃO

Nesta sequência é possível observar uma pessoa em ambiente externo (*outdoor*), em que acontecem várias movimentações da câmera, mudança da aparência do alvo (rotação) e também translações. Durante o processo de rastreamento existe uma leve similaridade de cores entre o fundo (*background*) do ambiente e as roupas da pessoa a ser rastreada. Na figura 5.1 são apresentados alguns *frames* da sequência. O quadrado branco mostra a estimativa do algoritmo proposto e na cor laranja as partículas representando o centro das regiões de amostragem.

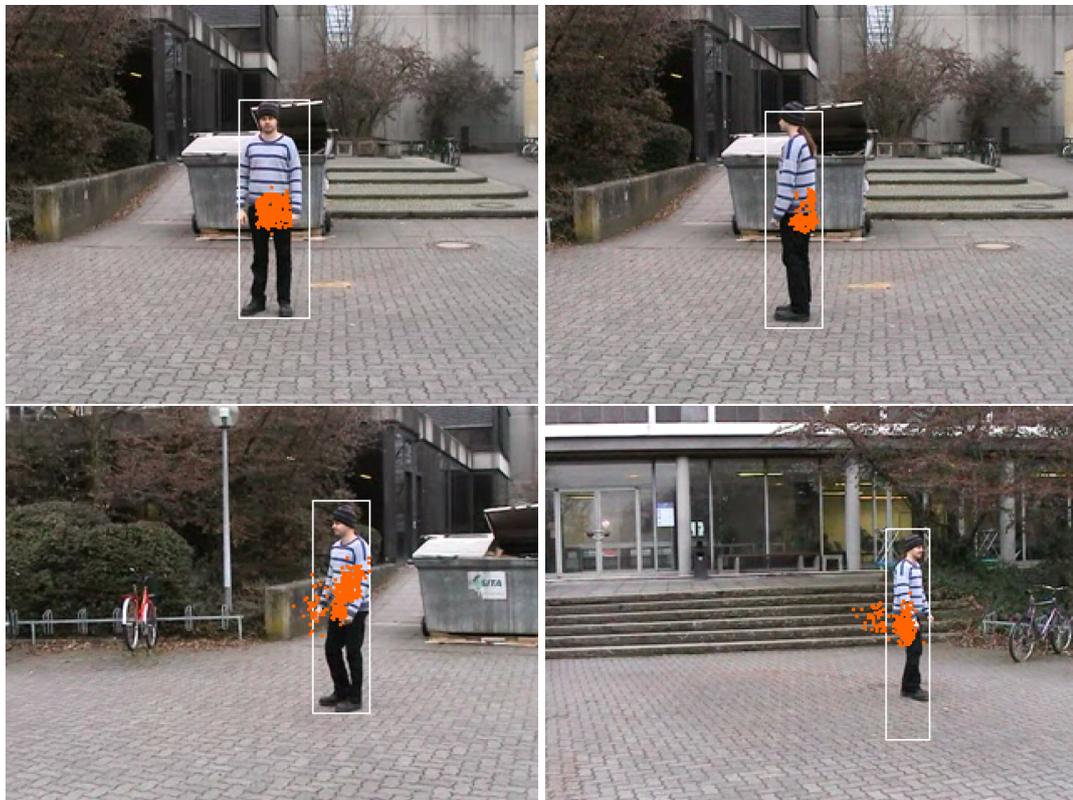


Figura 5.1: Resultado do rastreamento simples sem oclusão.

Com base nos dados fornecidos pelo *Ground Truth* e os obtidos pelo algoritmo proposto, pode-se fazer avaliações sobre a precisão do rastreamento realizado, para isso calcula-se o erro médio quadrático da diferença entre o valor fornecido pelo *Ground Truth* e aquele estimado pelo algoritmo proposto. Nos primeiros dois gráficos da Figura 5.2, tem-se a posição do centroide do quadrado contedor do objeto, normalizada a 1 para as coordenadas  $x$  e  $y$  em relação aos números de *frames*. Verifica-se que na coordenada  $x$  o filtro rastreia o objeto com um erro constante e baixo com comportamento consistente. Na coordenada em  $y$  é observado um comportamento com maior erro, entretanto o algoritmo consegue rastrear o

movimento vertical da pessoa com relativa precisão (em torno de 6%), constatado facilmente nas Figuras 5.2-(c) e (d) para as coordenadas  $x$  e  $y$  respectivamente.

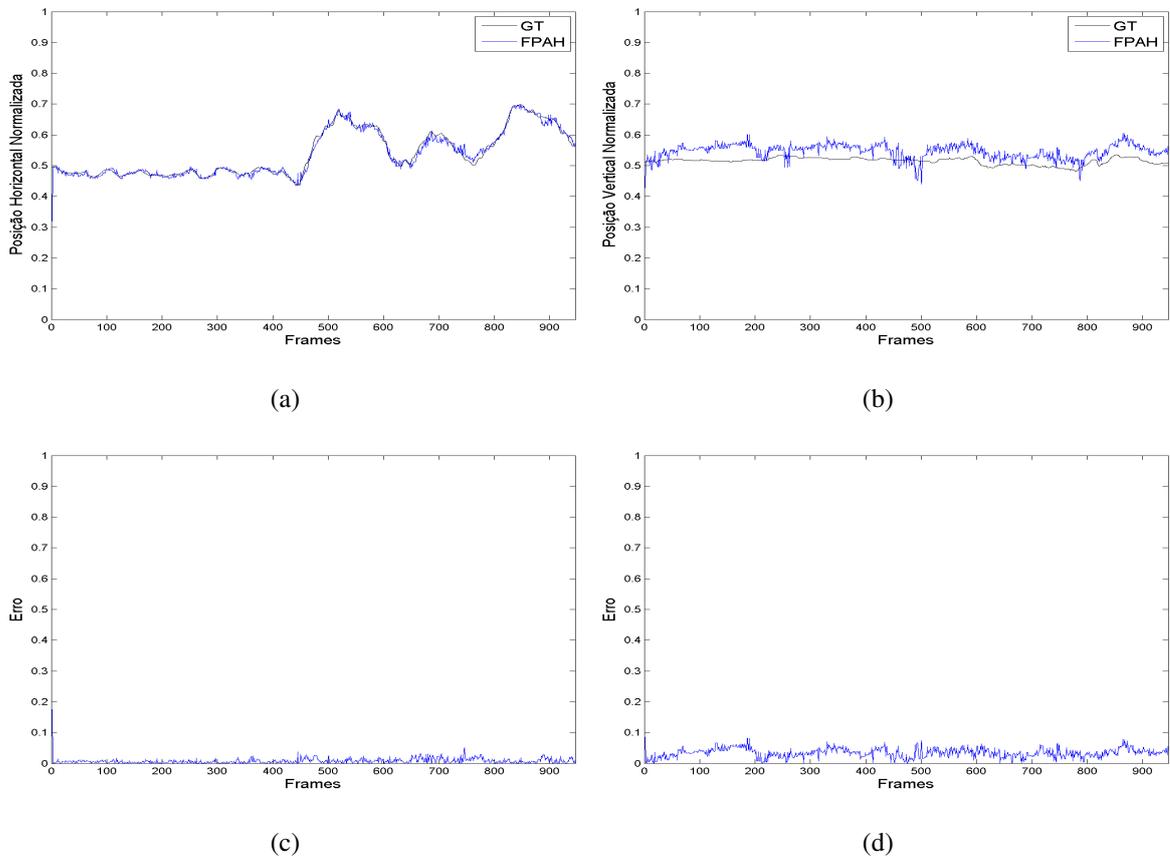


Figura 5.2: Sequência de rastreamento sem oclusão. Posições Normalizadas Horizontal (a) e Vertical (b) descritas pelo *Ground truth* em relação à metodologia proposta do FPAH. Erro da posição Horizontal (c) e Vertical (d).

## 5.2 RASTREAMENTO COM VARIAÇÃO DE ESCALA ESPACIAL

Nesta sequência (Figura 5.3) é realizado o rastreamento de um *rubik*, com diversas aproximações do observador gerando mudanças de escala espacial e também o ponto de vista. Nesta cena também são encontrados elementos visuais de distração com cores semelhantes ao objeto a ser rastreado.

No gráfico que representa o movimento na direção vertical (Figura 5.4-(b)), pode-se observar uma parte em que o centro da região de rastreamento se afasta do objeto (cubo). É facilmente percebido pela diferença entre a trajetória rastreada e as informações proveniente do *ground truth*. A causa principal deste fenômeno é distorção entre os pixels de fundo, presentes na escolha do alvo no *frame* inicial. Desta feita, o algoritmo tenta compensar a diferença entre as imagens dos frames, fazendo com que o rastreamento seja bem sucedido.

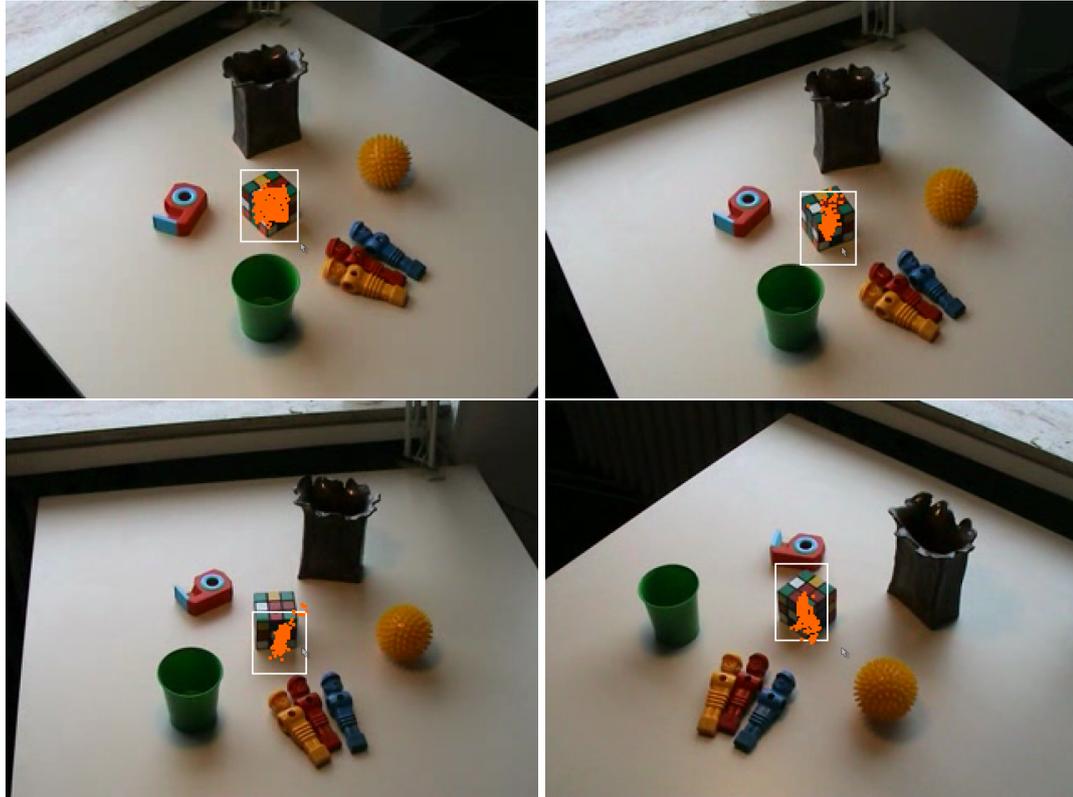


Figura 5.3: Sequência com mudança de escala espacial.

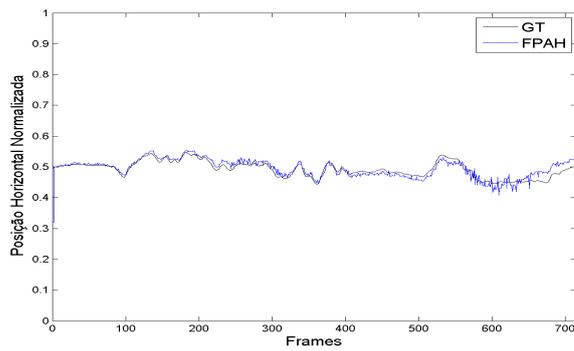
### 5.3 RASTREAMENTO COM VARIAÇÃO DE TRAJETÓRIA E VELOCIDADE DO OBJETO

Na Figura 5.5 é apresentada uma sequência em que uma bola de futebol gira, enquanto duas pessoas a levam de um lado para o outro. Nesta situação ocorre movimento da câmera e do alvo rotacionando e rápidas mudanças de trajetória do objeto.

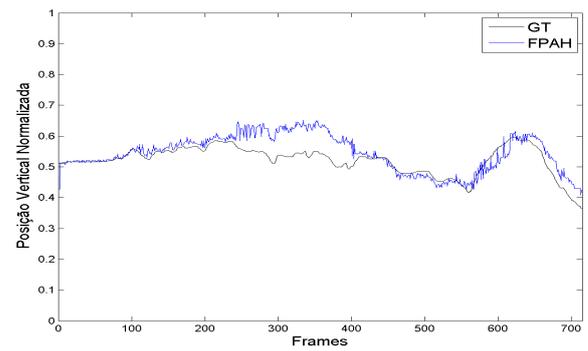
Nota-se pelos gráficos de movimento horizontal e vertical (Figura 5.6- (a) e (b)), que em alguns instantes o algoritmo proposto perde o alvo, quando apresentam-se variações rápidas de trajetória e velocidade da bola, entretanto a partir da abordagem de reamostragem híbrida com a inércia das partículas, rapidamente é recuperado o alvo seguindo todo o seu movimento. Os instantes em que ocorrem a perda do alvo são representadas pelos "picos" nos gráficos de Erro Normalizado (Figuras 5.6 - (c) e (d)) nas direções  $x$  e  $y$  respectivamente.

### 5.4 RASTREAMENTO COM VARIAÇÃO DE *BACKGROUND*

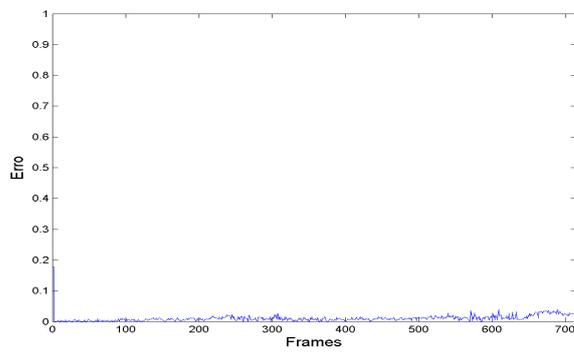
Nesta sequência (Figura 5.7) é possível observar as mudanças do *background* durante o rastreamento, juntamente com o movimento da câmera e do alvo, ainda com mudanças de escala. A cena é descrita por uma pessoa movendo um copo entre diferentes objetos de cor similar ao alvo do rastreamento.



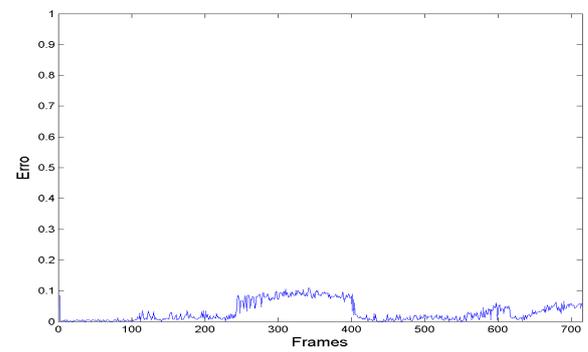
(a)



(b)



(c)



(d)

Figura 5.4: Sequência com mudança de escala espacial. Posições Normalizadas Horizontal(a) e Vertical (b) descritas pelo *Ground truth* e pelo **FPAH**. Erro da posição horizontal (c) e Vertical(d).

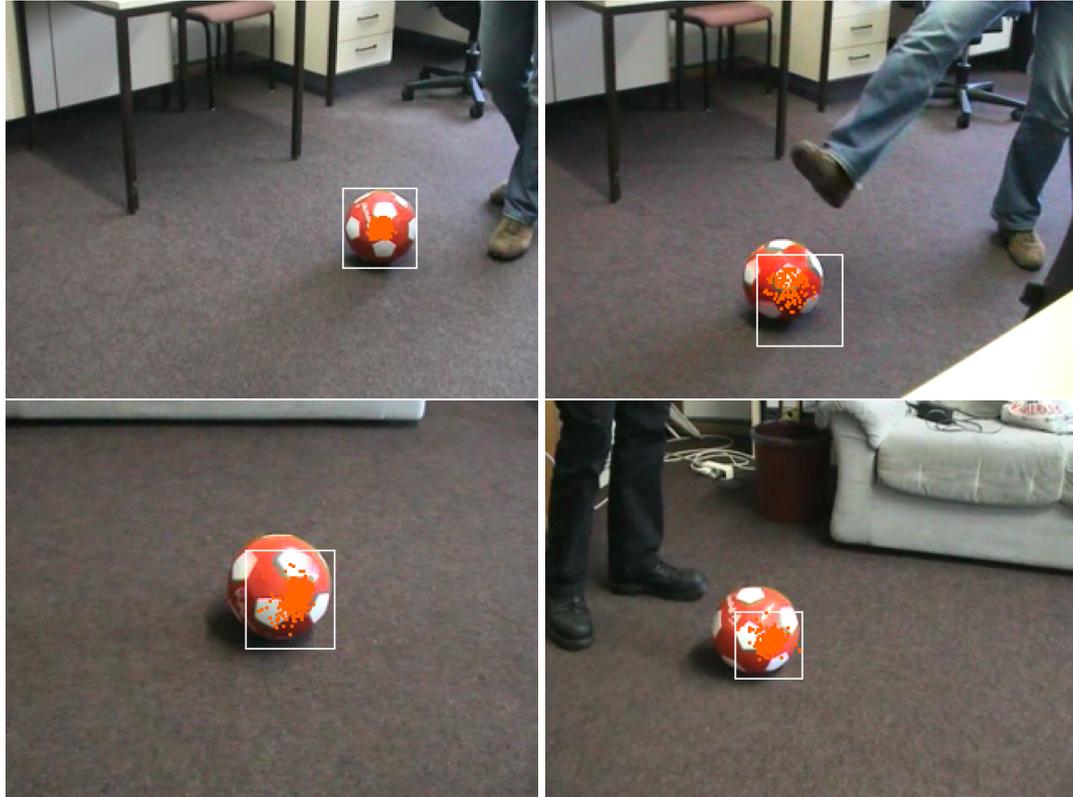


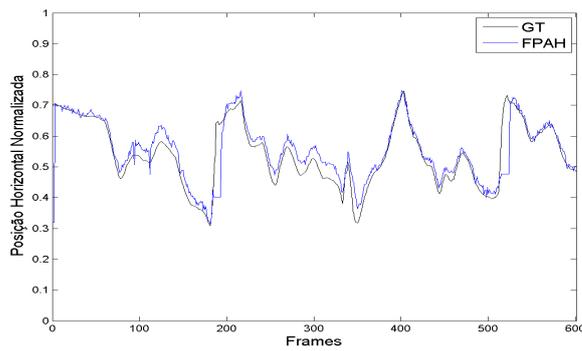
Figura 5.5: Sequência com mudanças rápidas de trajetória e velocidade.

Analisando os resultados do rastreamento (Figura 5.8), foi possível verificar que o algoritmo consegue rastrear durante toda execução da sequência o alvo com um erro muito pequeno. O valor do erro esteve inferior à 0,5 durante a maior parte da sequência (vide Figuras 5.8-(c) e (d)).

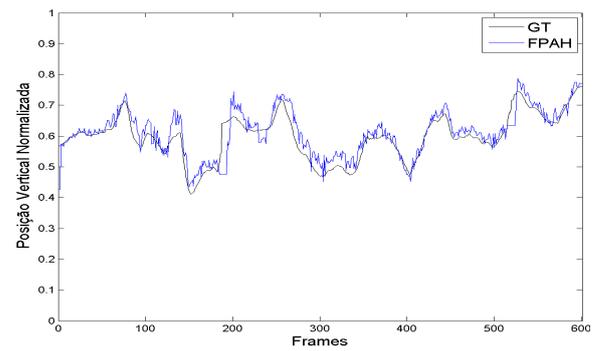
## 5.5 RASTREAMENTO COM VARIAÇÃO DE ILUMINAÇÃO

Na sequência utilizada para testes (Figura 5.9), são apresentadas mudanças de iluminação do ambiente a partir do controle da entrada de luz externa (via persiana ao fundo) em taxas variáveis. Além da entrada de luz externa, durante o decorrer da sequência utiliza-se uma fonte de iluminação artificial interna (lâmpada), alterando ainda mais a percepção dos objetos na cena. Para conseguir um bom desempenho do algoritmo frente a estas mudanças, o modelo de histograma a ser rastreado é atualizado durante a sequência seguindo o apresentado na Seção 4.

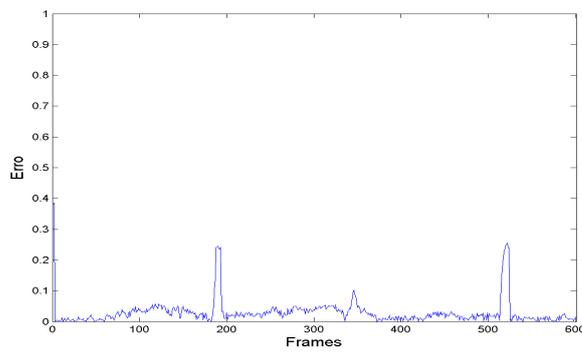
Nos gráficos relativos à posição do alvo apresentados na Figura 5.10 pode-se observar que as trajetórias do Ground Truth e a estimada pelo algoritmo proposto não são muito diferentes. Em alguns frames apresentam-se pequenas desviações por parte da estimação do método proposto, isto é causado nos momentos de mudança da iluminação, principalmente porque a atualização do histograma esta afetado com um pequeno ruído na estabilização



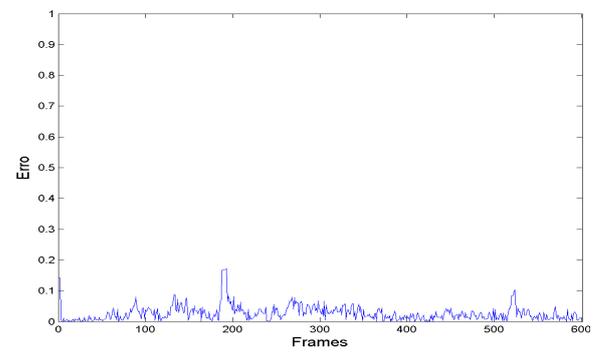
(a)



(b)



(c)



(d)

Figura 5.6: Sequência com mudança de trajetória. Posições Normalizadas Horizontal (a) e Vertical (b) descritas pelo *Ground truth* e pelo FPAH. Erro da posição Horizontal (c) e Vertical (d).

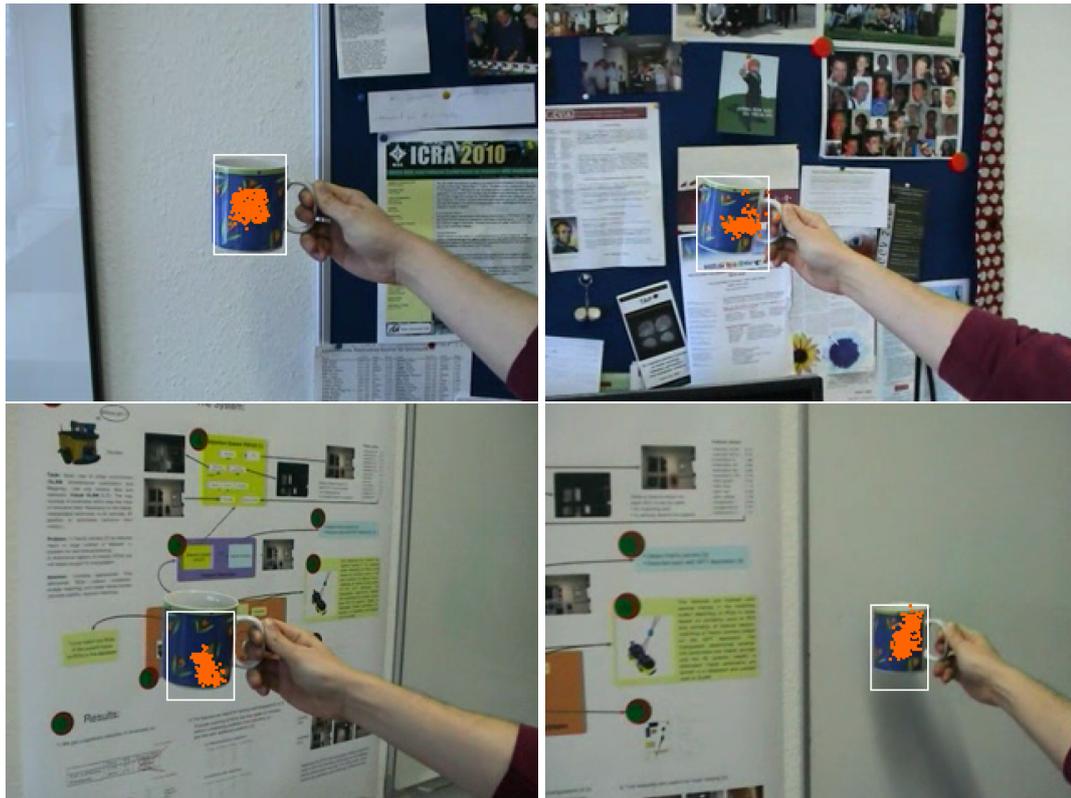


Figura 5.7: Sequência com variação do *background*.

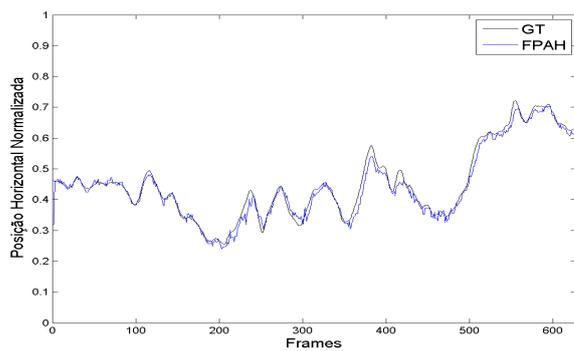
do modelo, mas a utilização da reamostragem com a inércia das partículas fazem que o rastreamento seja eficiente e o modelo do histograma seja atualizado de maneira correta.

## 5.6 RASTREAMENTO COM OCLUSÕES PARCIAIS E TOTAIS

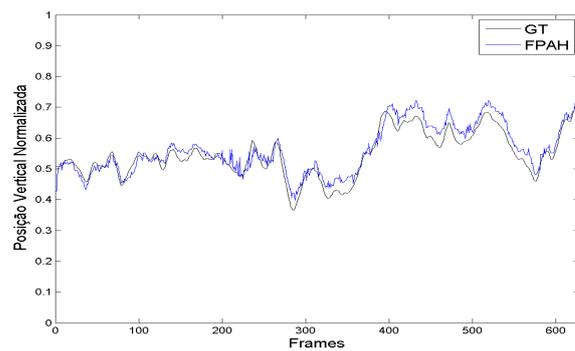
Situações envolvendo oclusões (sejam parciais ou totais) fazem parte da motivação para o desenvolvimento da metodologia proposta. Os transtornos causados pelo fenômeno de oclusão (parcial ou total) do alvo no rastreamento visual são inúmeros, podendo interferir fortemente em aplicações em que o rastreamento do alvo é parte fundamental do processo (ex.: aplicações em controle servo-visual).

A sequência apresentada na Figura 5.11, observa-se uma pessoa ao lado de uma coluna e a partir do movimento da câmera ocorre uma oclusão parcial do alvo (pessoa atrás da coluna). Este movimento seguido pelo retorno da câmera à posição original.

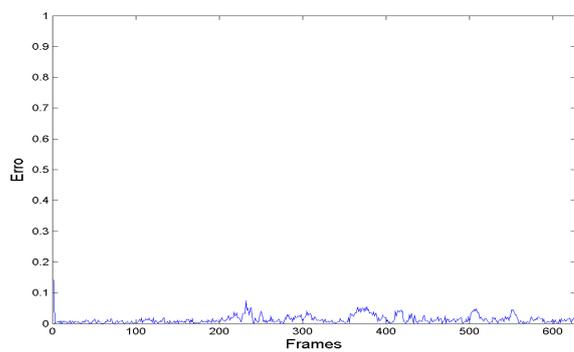
Nos gráficos que descrevem todo o rastreamento realizado (Figura 5.12), pode-se observar que a metodologia proposta obteve sucesso. A analisar os resultados do rastreamento a ocorrência de uma pequena variação, em relação ao *ground truth*, da posição vertical estimada. Este fenômeno ocorre devido à baixa informação do *background* presente no modelo inicial do alvo a ser rastreado. Quando ocorre a oclusão do objeto por detrás da coluna, a me-



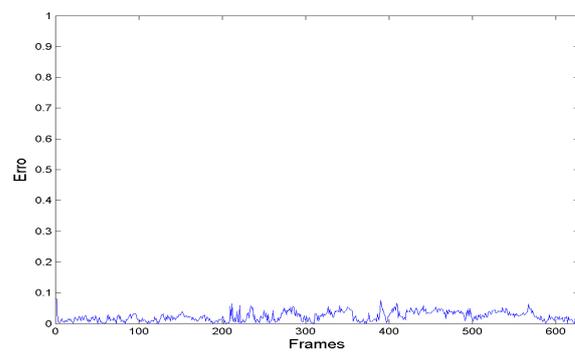
(a)



(b)



(c)



(d)

Figura 5.8: Sequência com variação do *background*. Posições Normalizadas Horizontal (a) e Vertical (b) descritas pelo *Ground truth* e pelo FPAH. Erro da posição Horizontal (c) e Vertical (d).

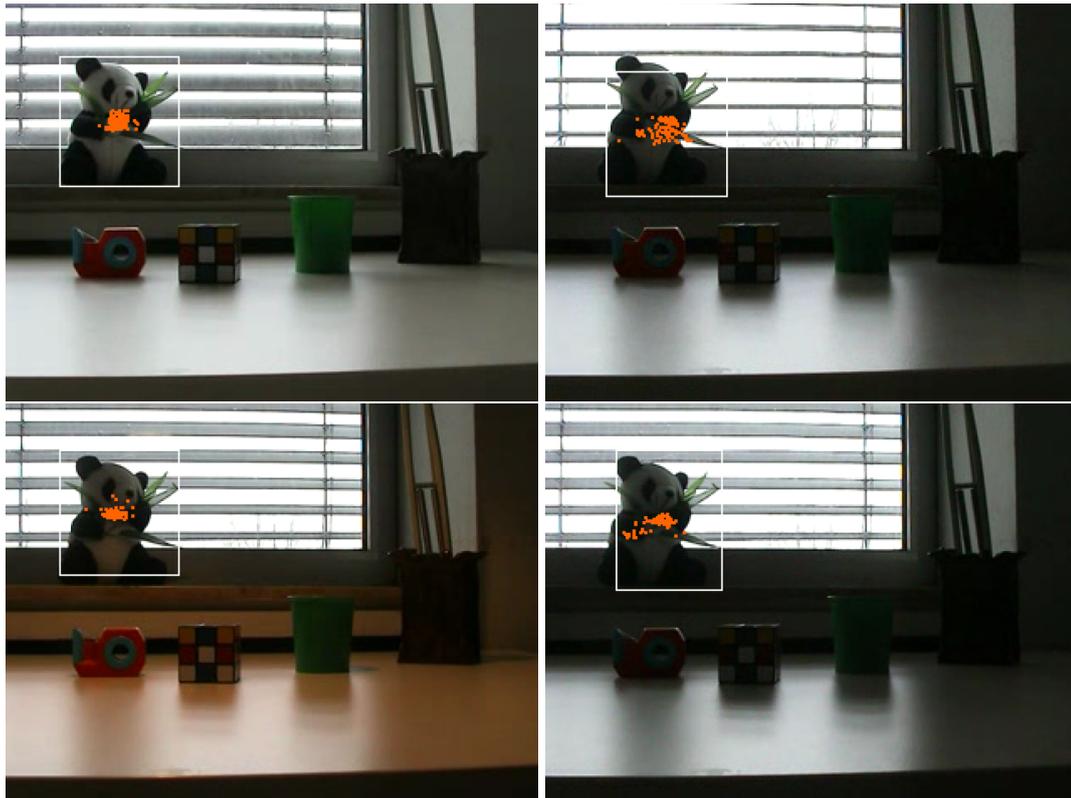
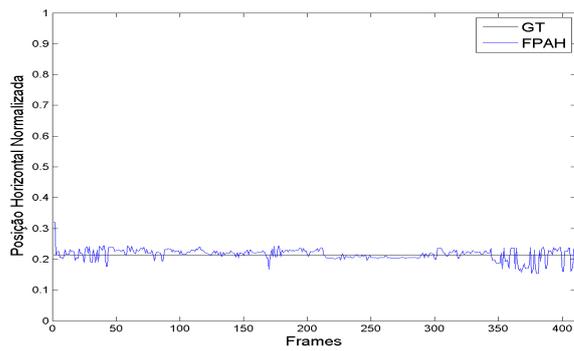


Figura 5.9: Sequência com mudanças de iluminação.

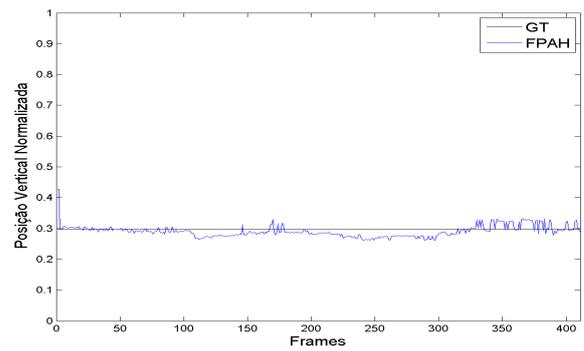
todo a metodologia proposta tenta compensar os efeitos causados por esta oclusão fazendo a estimação da posição mediante a posição da melhor partícula.

Entretanto na Figura 5.13 a sequência mostra uma pessoa que caminha num corredor, que durante seu movimento as colunas se interpõem entre a câmera e a pessoa, originando a oclusão total. Ainda durante a sequência outra pessoa realiza a oclusão do alvo a ser rastreado (pessoa com blusa cinza com listras finas) utilizando roupas com cores similares (pessoa com blusa cinza com listras grossas) a do alvo. Antes da oclusão as partículas em cor laranja estão próximas ao objeto. No momento da oclusão, estas partículas mudam para a cor cinza indicando a oclusão. Observando a dispersão das partículas no momento da oclusão verifica-se uma forma achatada da distribuição de partículas, indicando que as partículas estão fortemente influenciadas pelas informações da inércia do movimento (ou seja, apontam na direção do movimento). No momento em que as partículas detectam novamente o alvo, o amostragem continua como no instante anterior a oclusão ocorrer, o modelo do histograma padrão volta a ser atualizado com as novas observações o que garante robustez para continuar com o rastreamento.

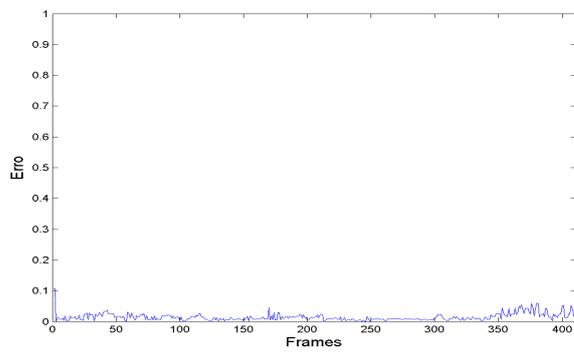
Nos gráficos do movimento vertical e horizontal da Figura 5.14, pode-se observar que a posição do centro da janela que envolve o alvo, obtido pela aplicação, esta bem próximo da posição do centro informado pelo *Ground Truth*. Quando ocorre uma oclusão, por convenção dos criadores do *benchmark* utilizado (vide [48]), o *Ground Truth* atribui por convenção o



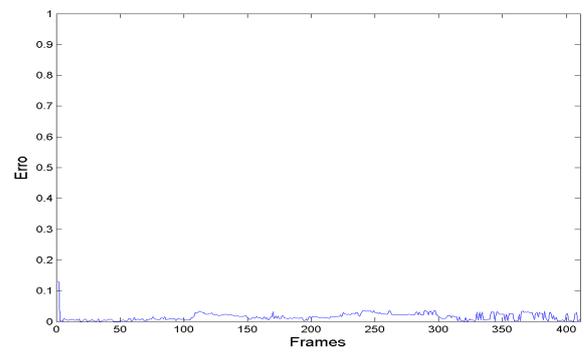
(a)



(b)



(c)



(d)

Figura 5.10: Sequência com mudanças de iluminação. Posições Normalizadas Horizontal (a) e Vertical (b) descritas pelo Ground truth e pelo FPAH. Erro da posição Horizontal (c) e Vertical (d)

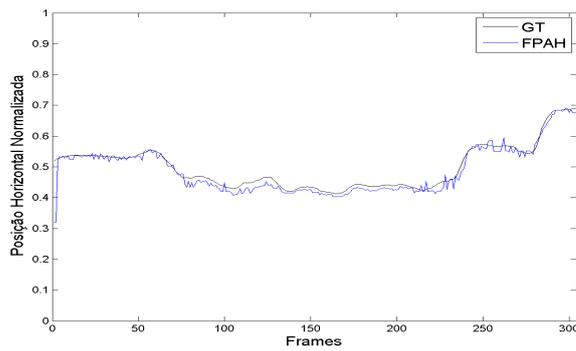


Figura 5.11: Sequência com oclusão parcial de uma pessoa.

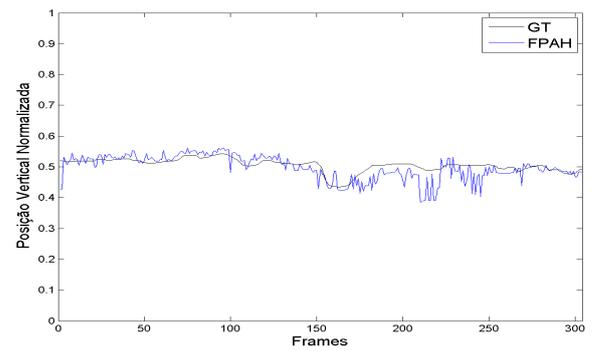
valor **zero** para todas as variáveis de posição do alvo. Mas o algoritmo proposto faz uma estimativa da posição, apesar da colusão efetuada pela coluna, o que gera valores de erro elevados nas Figuras 5.14 - (c) e (d), respectivamente.

Na Figura 5.15 pode-se observar as imagens de uma sequência em que uma pessoa caminha num ambiente externo (*outdoor*). Durante o movimento outras pessoa sobrepõem o alvo (pessoa da casaco preto) gerando diversas oclusões totais e parciais. Durante a sequência a pessoa, diga-se o alvo, se movimenta com variação de escala e rotação. Neste caso, as partículas atualizadas no processo de reamostragem inercial são alimentadas pela informação do movimento durante a oclusão. As demais permanecem fixas e são encarregadas em detectar o alvo pela melhor partícula ao final da oclusão.

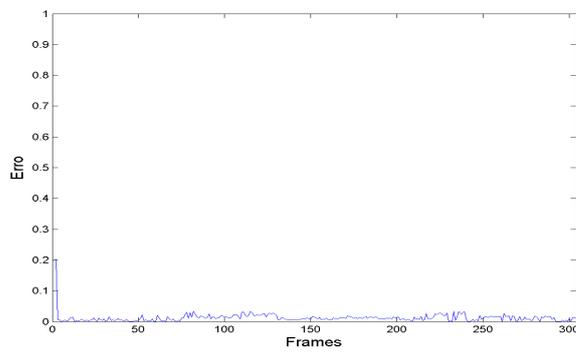
A partir da análise dos gráficos que representam os movimentos horizontal e vertical do alvo (Figura 5.16), é possível observar o rastreamento obtido antes e depois de ocorrida uma oclusão. É possível também verificar que o rastreamento foi realizado com sucesso, e ainda que a metodologia proposta alcançou um nível de robustez elevado para situações com oclusões. Os valores para os erros em relação ao *ground truth* oscilaram entre 0, 2 durante a sequência realizada, salvo os instantes em que ocorre a oclusão total. Neste caso, os valores de erro foram elevados devido à convenção do *benchmark* utilizado colocar o valor zero para a posição do alvo na imagem. É importante também ressaltar que a trajetória estimada pela metodologia proposta, mesmo com pequenas oscilações, descreve uma trajetória esperado



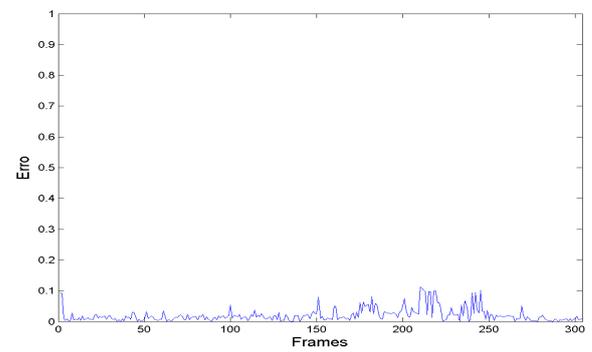
(a)



(b)



(c)



(d)

Figura 5.12: Sequência como oclusão parcial. Posições Normalizadas Horizontal (a) e Vertical (b) descritas pelo Ground truth e pelo FPAH. Erro da posição Horizontal (c) e Vertical (d).

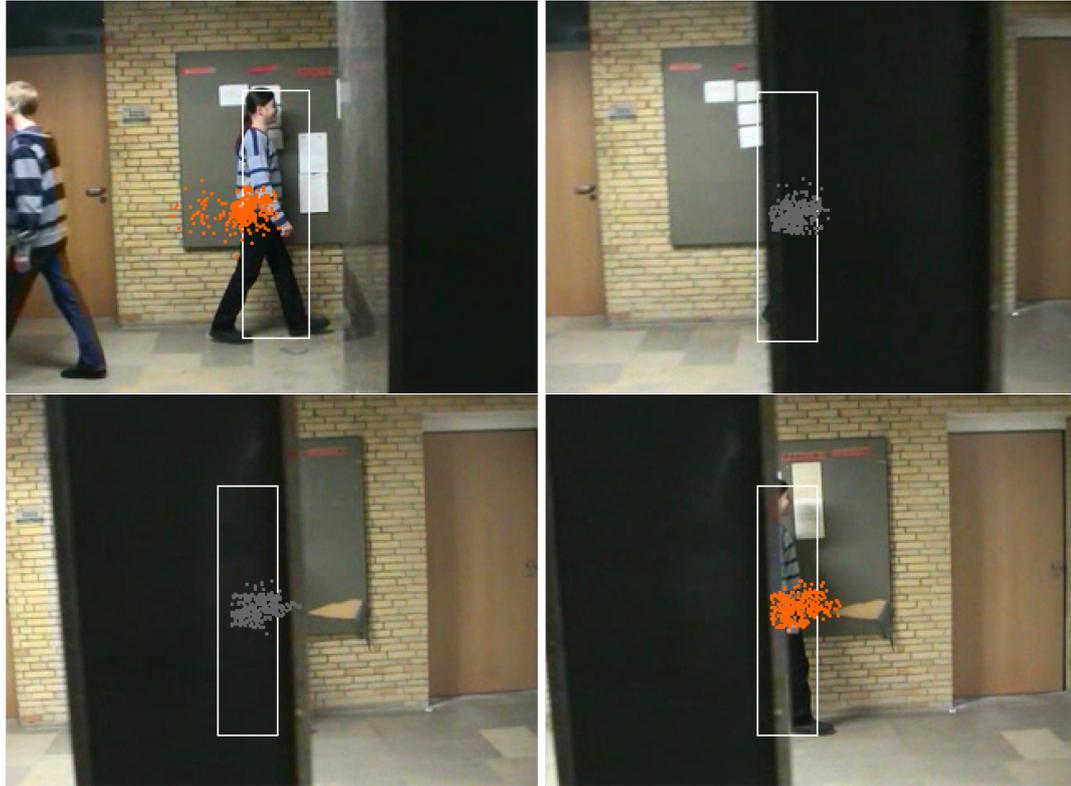


Figura 5.13: Sequência com oclusão total.

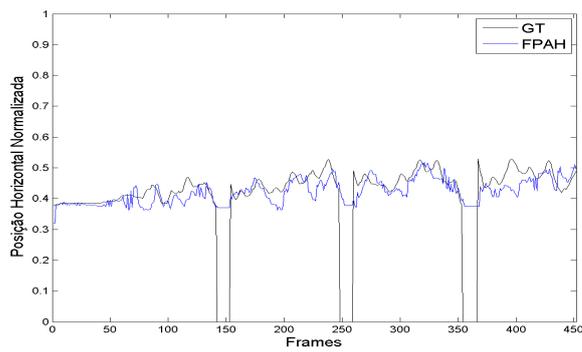
para o alvo a ser rastreado.

## 5.7 RASTREAMENTO PARA MÚLTIPLOS ALVOS

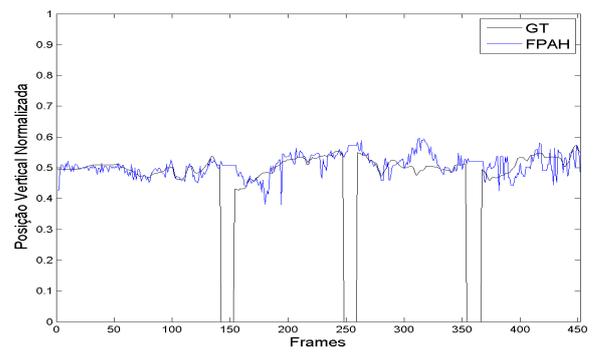
Com intuito de testar não somente a robustez da metodologia proposta, mas também sua flexibilidade para diversas aplicações, foram realizados testes para rastreamento de vários alvos. Na Figura 5.17 é descrita uma sequência onde duas pessoas caminham por um corredor cruzando suas trajetórias na imagem, por fim originando diversas oclusões. Também nesta sequência é notada mudanças de iluminação geradas pela disposição das fontes de iluminação no corredor.

As configurações do filtro FPAH, referentes ao número e distribuição de partículas, foram as mesmas utilizadas nas Seções anteriores. Devido ao aumento da carga computacional do rastreamento, foi verificada um aumento no tempo de processamento do rastreamento. Infelizmente os valores mensurados não são adequados para comparações efetivas devido ao sistema operacional utilizado não ter sido configurado para atender às demandas de medição de tempo apropriadas.

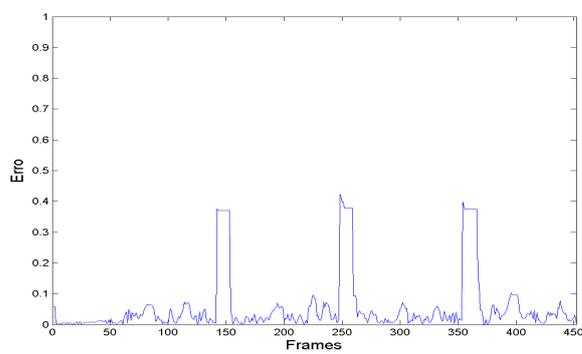
Contudo, o rastreamento foi realizado corretamente (vide Figura 5.18) em relação ao *ground truth* e os valores do erro apresentados nas Figuras 5.18 - (c) e (d), respectivamente,



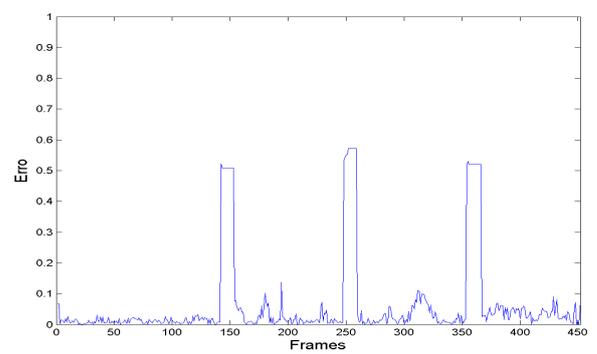
(a)



(b)



(c)



(d)

Figura 5.14: Sequência com oclusão total. Posições Normalizadas Horizontal (a) e Vertical (b) descritas pelo Ground truth e pelo FPAH. Erro da posição Horizontal (c) e Vertical (d).

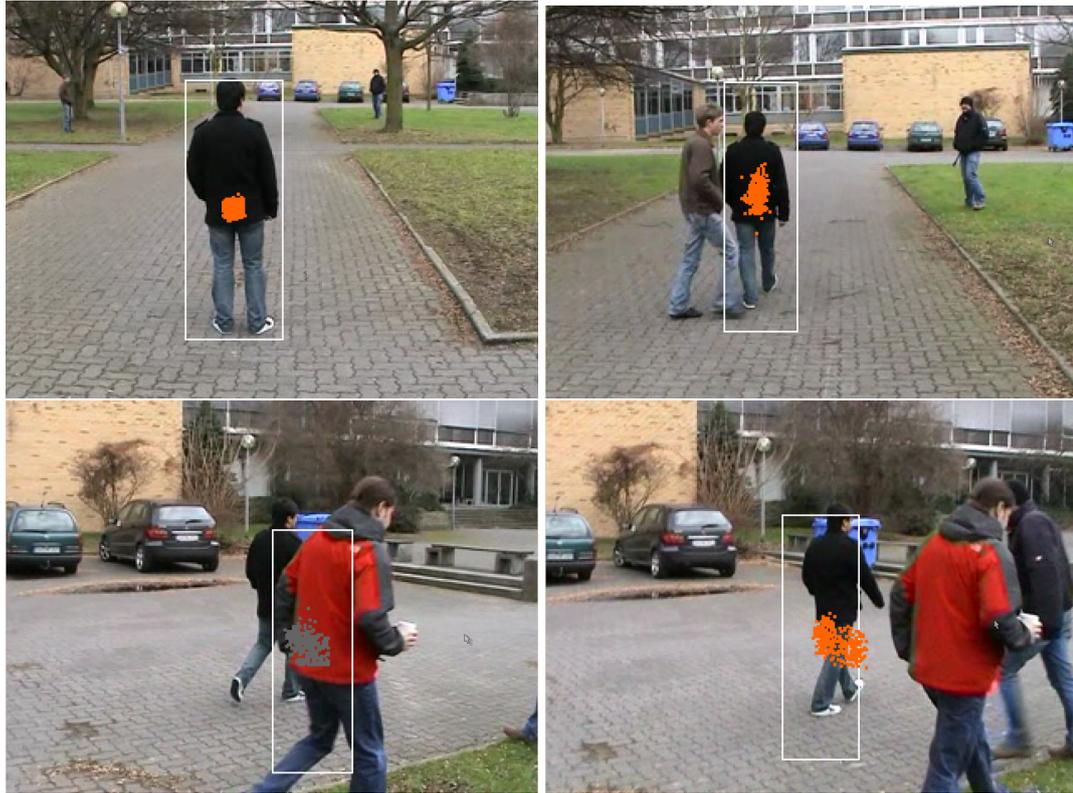


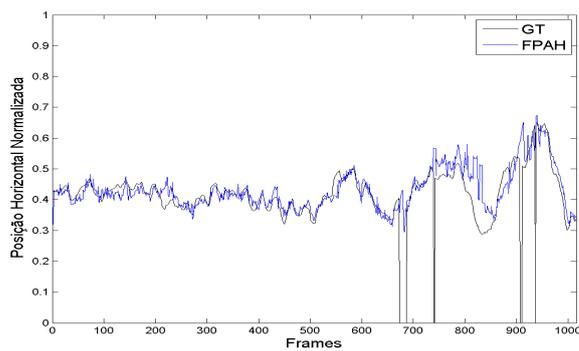
Figura 5.15: Sequência em ambiente externo com oclusão total.

em que os valores elevados são ocorridos devido à convenção do *Benchmark* colocar valor zero quando ocorrem oclusões.

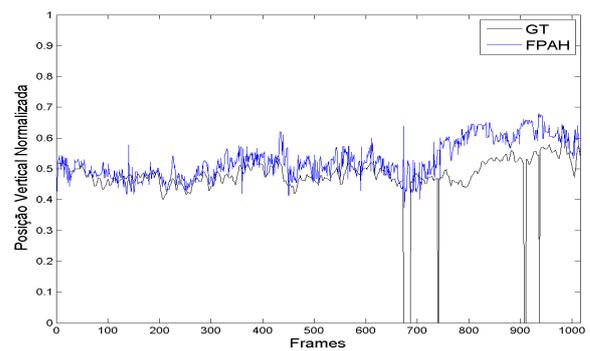
Para esta sequência, o filtro FPAH utilizou um total de 200 partículas para cada uma das pessoas, distribuindo a reamostragem seguindo as mesmas proporções do apresentado inicialmente. Vale ressaltar que o custo computacional eleva-se sensivelmente, pois são lançados dois processos de rastreamento independentes.

## 5.8 COMPARATIVO ENTRE ALGORITMOS DE RASTREAMENTO E A METODOLOGIA PROPOSTA

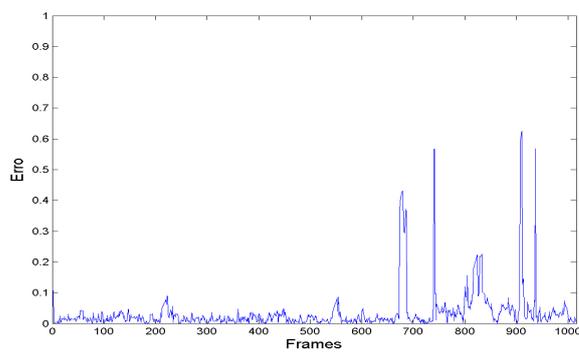
Nesta seção são abordados os aspectos comparativos de robustez, e até mesmo de precisão da metodologia proposta. Os testes foram conduzidos utilizando os seguintes algoritmos de rastreamento: o algoritmo determinístico *Template/Window Matching (WM)* baseado no rastreamento por correspondência de regiões[25], e com o algoritmo probabilístico de rastreamento do filtro de partículas com amostragem por reamostragem de importância (SIR), o qual tem o mesmo número de partículas e condições de desvio padrão para cada uma das distribuições empregadas tal qual o Filtro de Partículas por amostragem de importância híbrida (FPAH) proposto.



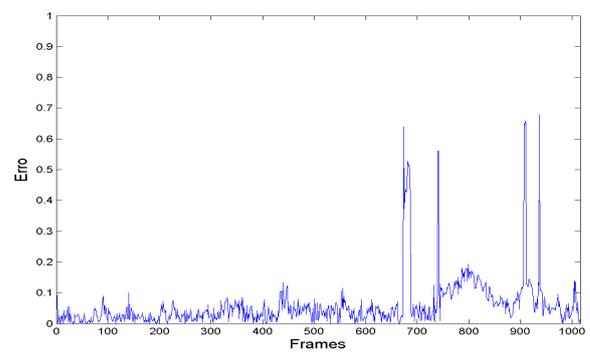
(a)



(b)



(c)



(d)

Figura 5.16: Sequência com oclusão *outdoor*. Posições Normalizadas Horizontal (a) e Vertical (b) descritas pelo Ground truth e pelo FPAH. Erro da posição Horizontal (c) e Vertical (d).



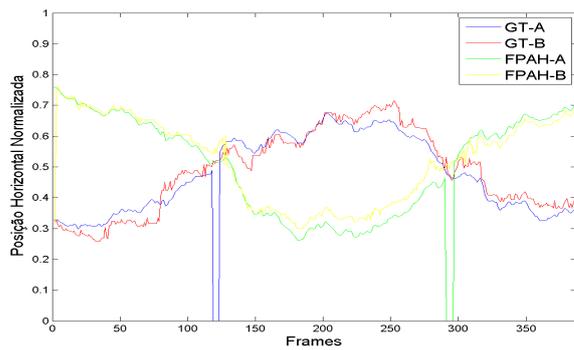
Figura 5.17: Sequência de duas pessoas caminhando por um corredor com oclusão total e mudanças de iluminação.

Para esta avaliação os algoritmos serão testados para as seguintes situações: *Rastreamento simples sem oclusão; com variação de escala espacial; variação de trajetória e velocidade do objeto; variação de background; variação de iluminação; e oclusões totais.*

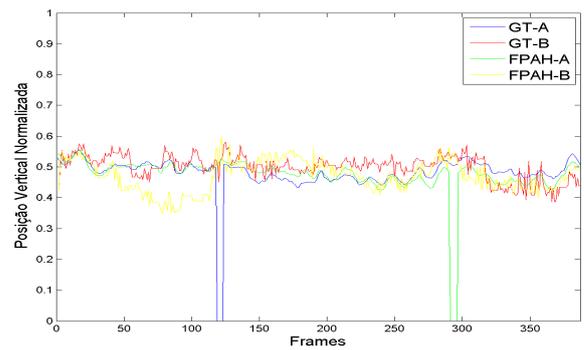
### 5.8.1 Rastreamento simples sem oclusão

A seguir (Figura 5.19) são apresentadas alguns *frames* que compõe a sequência. Esta sequência descreve uma pessoa em um ambiente externo (*outdoor*) descrevendo movimentos simples. Verifica-se que o algoritmo *WM* não consegue realizar o rastreamento com sucesso, principalmente após serem realizadas mudanças de escala do alvo. O algoritmo do filtro *SIR* consegue realizar o rastreamento com sucesso. Entretanto, após ao final do rastreamento a área rastreado tem sua dimensões alteradas, comprometendo a localização exata do alvo. Para a metodologia proposta (algoritmo *FPAH*), esta mantém sua estimativa da posição do alvo, bem como as estimativas corretas para as dimensões do alvo durante o rastreamento, permitindo sua correta localização na imagem.

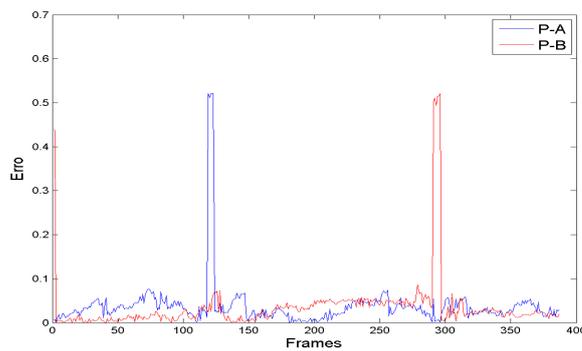
Nos gráficos da Figura 5.20, no momento que ocorre a mudança de escala do alvo a ser rastreado, o algoritmo *WM* perde o alvo devido as grandes variações no movimento horizontal e vertical. Analisando o comportamento do filtro *SIR*, mesmo que o rastreamento tenha sido realizado com sucesso, ocorrem variações bruscas de estimações de posições, o que



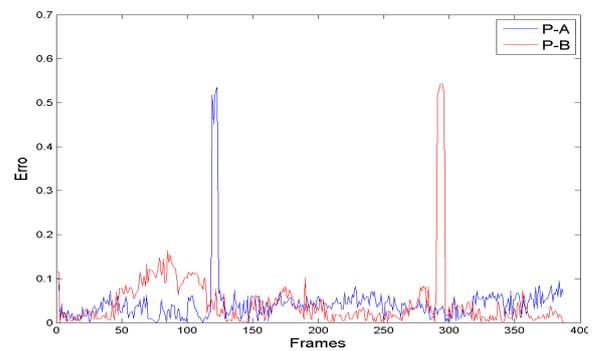
(a)



(b)



(c)



(d)

Figura 5.18: Sequência de rastreamento para múltiplos alvos. Posições Normalizadas horizontal (a) e vertical (b) descritas pelo *ground truth* para a pessoa A **GT-A** e para a pessoa B **GT-B** e pelo FPAH estimado para pessoa **FPAH-A** e pessoa B **FPAH-A**. Erro da estimação da posição para pessoas **A**, **B** nas orientações Horizontal(c) e Vertical(d).

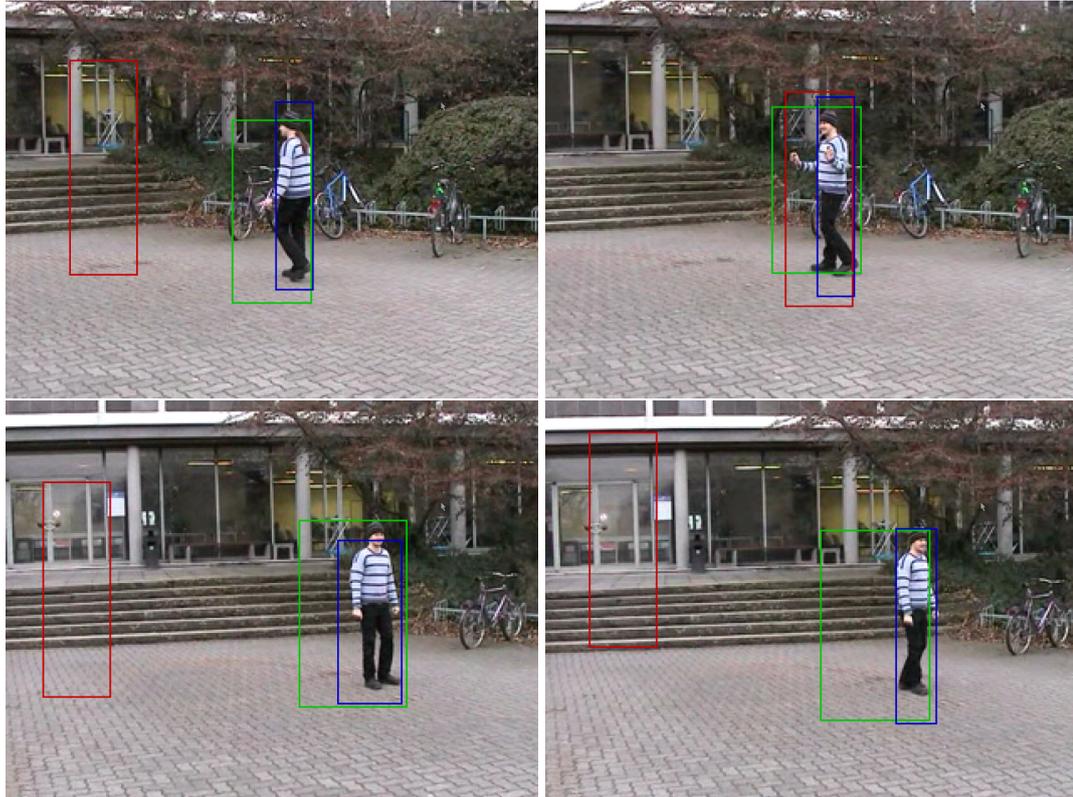


Figura 5.19: Desempenho dos algoritmos seqüência sem oclusão: **WM**, **SIR** e **FPAH**.

acaba prejudicando a precisão de todo o processo. Com o filtro FPAH, facilmente verifica-se que este acompanha o alvo durante toda a seqüência, mantendo as estimativas de dimensões do alvo e descrevendo uma trajetória mais suave de acordo com o padrão de movimento fornecido pelo *Ground Truth*.

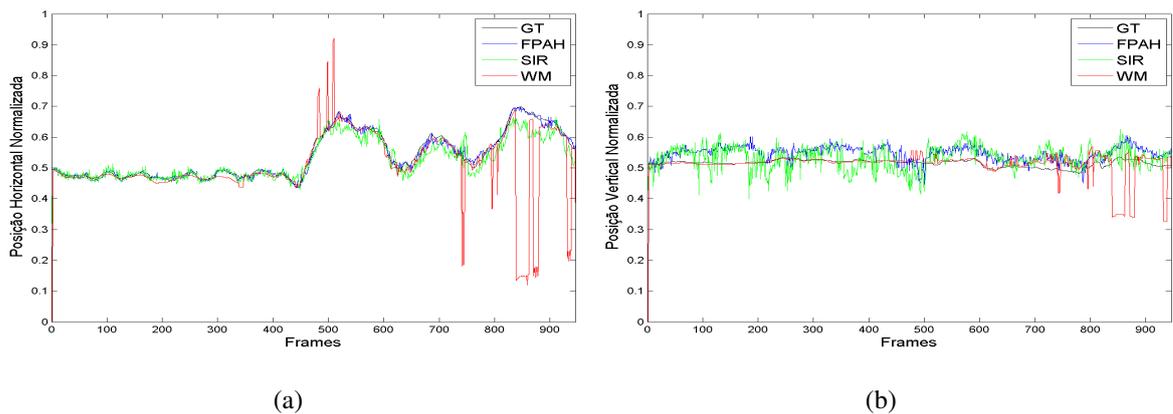


Figura 5.20: Seqüência sem oclusão. *Ground Truth* e estimaco da posico dos algoritmos **WM**, **SIR** e **FPAH** orientaco Horizontal(a) e Vertical(b).

### 5.8.2 Rastreamento com variação de escala espacial

Nesta sequência o alvo é um cubo colorido que está sobre a mesa, ao lado de diversos objetos com cores semelhantes. Os resultados do rastreamento realizado é apresentado na Figura 5.21. Avaliando os resultados obtidos, nota-se que no momento de mudanças de escala do objeto devido ao movimento da câmera, o algoritmo *WM* não consegue realizar o rastreamento corretamente. O filtro *SIR* consegue rastrear o cubo, mas novamente não consegue estimar corretamente a escala das dimensões do alvo. E também, novamente, o filtro *FPAH* realiza todo o rastreamento o cubo, mantendo a trajetória suave e mantém sua escala na maioria dos quadros.

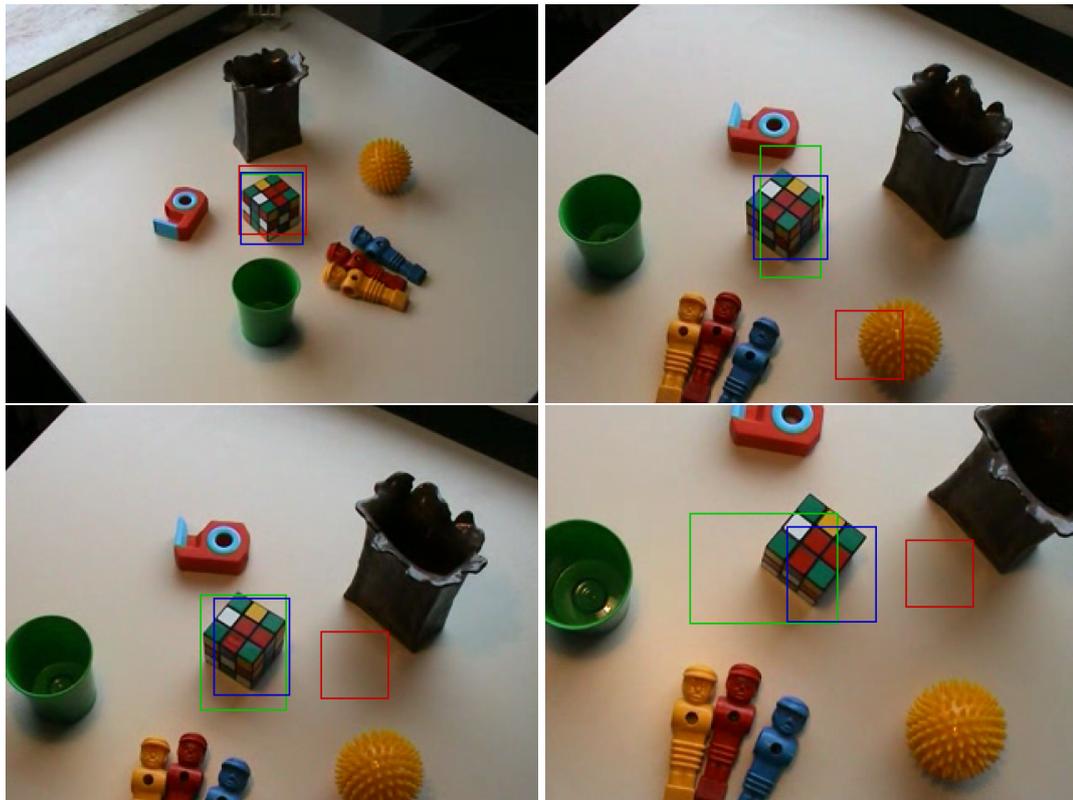


Figura 5.21: Sequência com variação de escala espacial: *WM*, *SIR* e *FPAH*.

Nos gráficos da Figura 5.22 são descritas as trajetórias realizadas durante o processo de rastreamento. É evidente os momentos de perda do alvo rastreado pelo algoritmo *WM*. A resposta dos filtros *SIR* e *FPAH* são similares e conseguem acompanhar o objeto com um pequeno erro na estimação da posição do centróide na direção vertical. No caso do algoritmo *FPAH*, no momento da mudança do deslocamento da posição da câmera, o algoritmo tenta compensar os pixels do fundo apresentados no modelo inicial do rastreamento, de maneira tal a manter estável todo o processo.

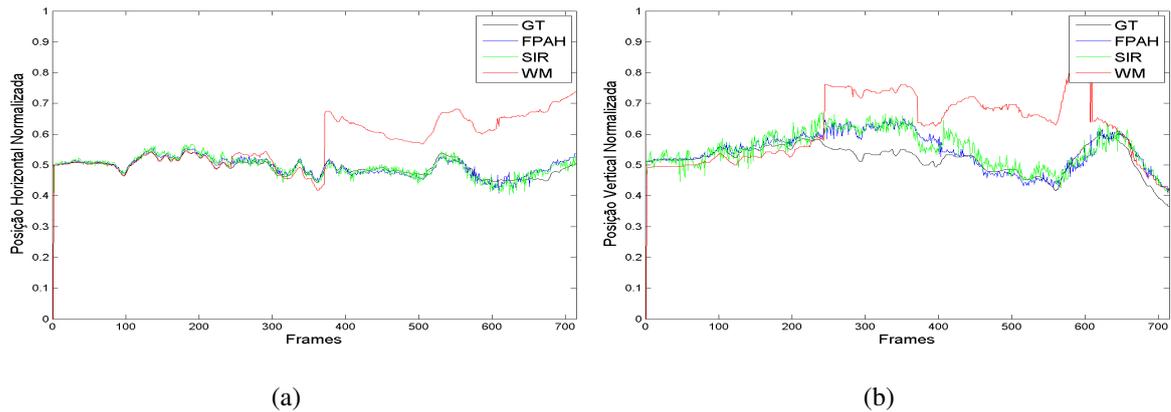


Figura 5.22: Sequência com variação de escala espacial. *Ground Truth* e estimação da posição dos algoritmos **WM**, **SIR** e **FPAH** nas direções Horizontal(a) e Vertical(b).

### 5.8.3 Rastreamento com variação de trajetória e velocidade do objeto

A sequência apresentada na Figura 5.23, também utilizada nos testes descritos na Seção 5.3, a trajetória realizada pelo alvo (bola laranja) durante toda a sequência de movimentos é composta de variações de escala e mudanças abruptas de velocidades e trajetórias.

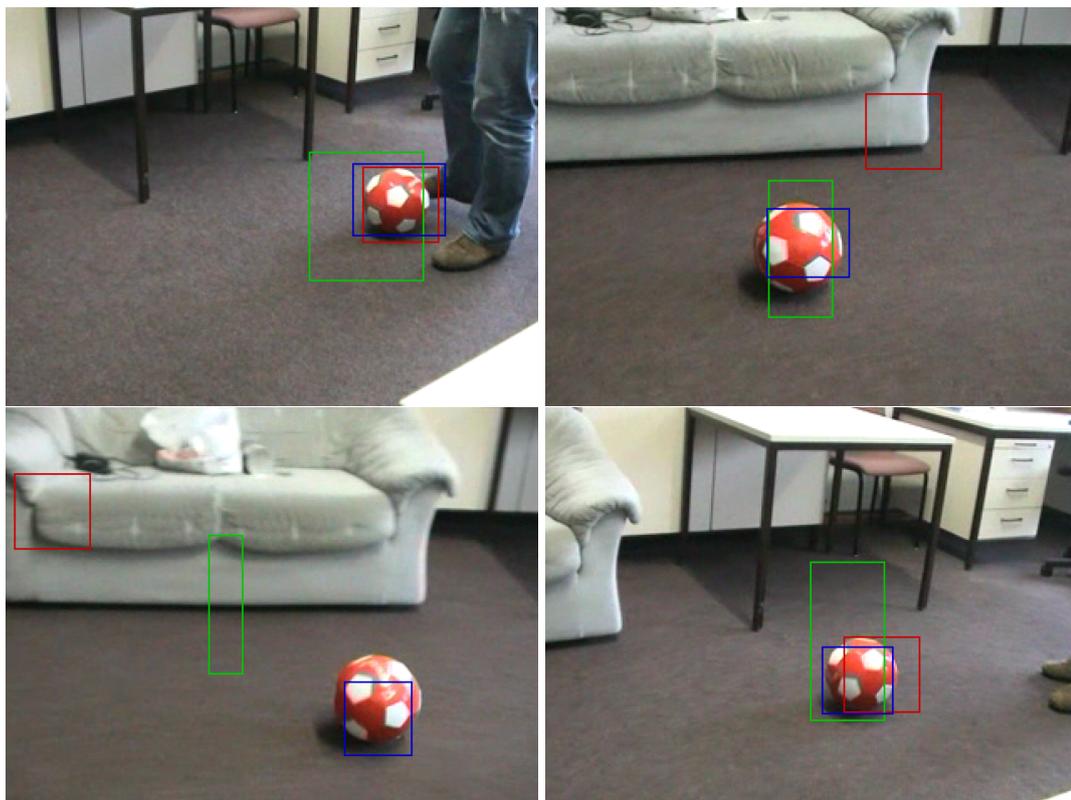


Figura 5.23: Sequência com variação de trajetória e velocidade do objeto: **WM**, **SIR** e **FPAH**.

Nas imagens da Figura 5.23, verifica-se a perda do alvo por parte dos algoritmos **WM** e do filtro **SIR** próximo ao instante em que a trajetória da bola muda. Ao contrário dos

demais, o filtro FPAH consegue recuperar rapidamente o rastreamento da bola durante as mudanças de trajetória. Isto ocorre devido a ativação do modelo inercial, o que permite o algoritmo manter o rastreamento após a alteração da trajetória. Nos gráficos de movimento horizontal e vertical respectivamente (Figura 5.24), percebe-se algumas diferenças entre o *Ground Truth* e as posições do *WM* e do filtro SIR. O resultado do algoritmo do filtro FPAH é mais consistente, alcançando o resultado esperado do rastreamento com algumas pequenas variações na estimativa da posição.

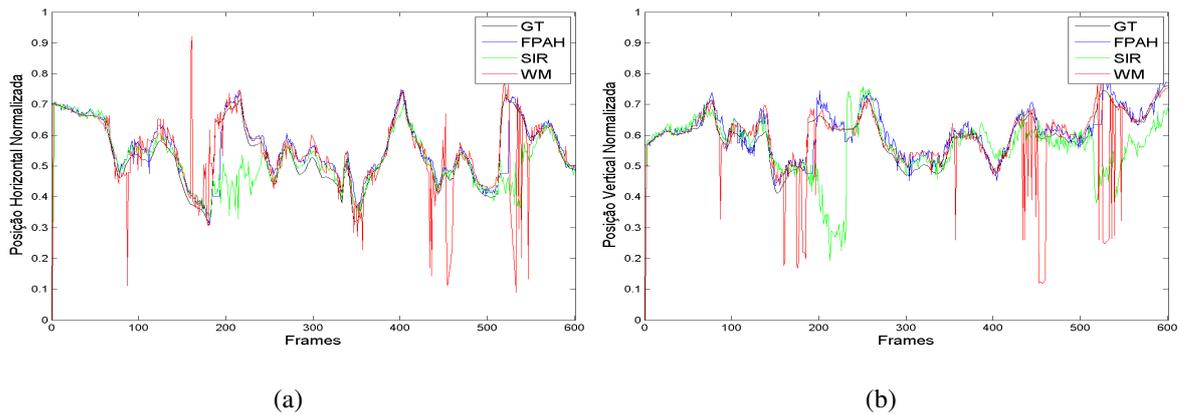


Figura 5.24: Sequência com variação de trajetória e velocidade do objeto. *Ground Truth* e posição dos algoritmos *WM*, *SIR* e *FPAH* nas direções Horizontal(a) e Vertical(b).

#### 5.8.4 Rastreamento com variação de *background*

Para padronizar a realização das comparações entre os algoritmos, utilizou-se a mesma sequência da Seção 5.4 em que um copo é movimentado sobre regiões com elevado grau de ambiguidade e mudanças de escala (Figura 5.25).

A resposta do algoritmo *WM* apresenta erros no rastreamento do copo, especialmente quando o fundo tem aparência fortemente similar ao copo. O filtro SIR tem uma boa resposta com relação a posição, mas sua escala de estimativa da dimensão do alvo se perde no decorrer da sequência. Ao avaliar a saída do filtro FPAH facilmente verifica-se que é consistente com relação a posição e a escala para cada uma das variações do copo na sequência. Na Figura 5.26 os gráficos das posições vertical e horizontal, destacam-se os erros de estimativa gerados por *WM*. A resposta do filtro SIR e do filtro FPAH são similares e acompanham o alvo com uma boa aproximação ao *ground truth*.

#### 5.8.5 Rastreamento com oclusão total

Nas imagens desta sequência (vide Figura 5.27), observa-se que o algoritmo *WM* não consegue rastrear o alvo corretamente, antes mesmo da ocorrência da oclusão, não conseguindo recuperar sua posição no restante da sequência. O filtro SIR consegue realizar o

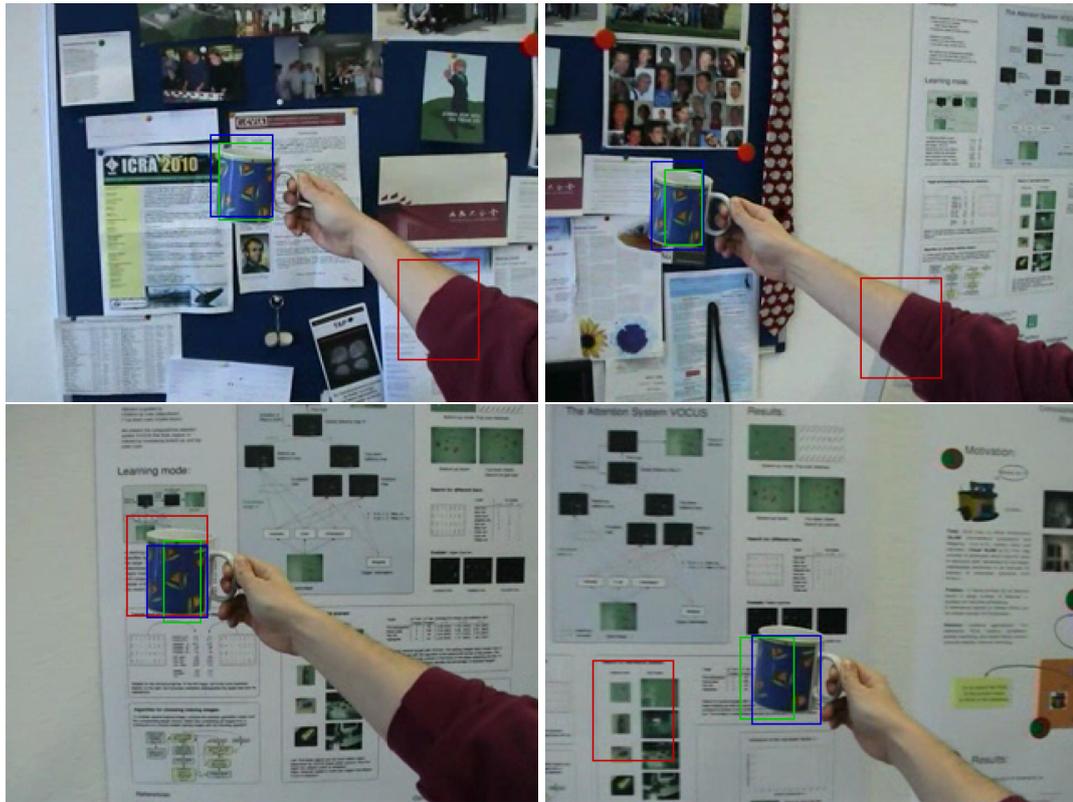


Figura 5.25: Sequência com variação de *background*: WM, SIR e FPAH.

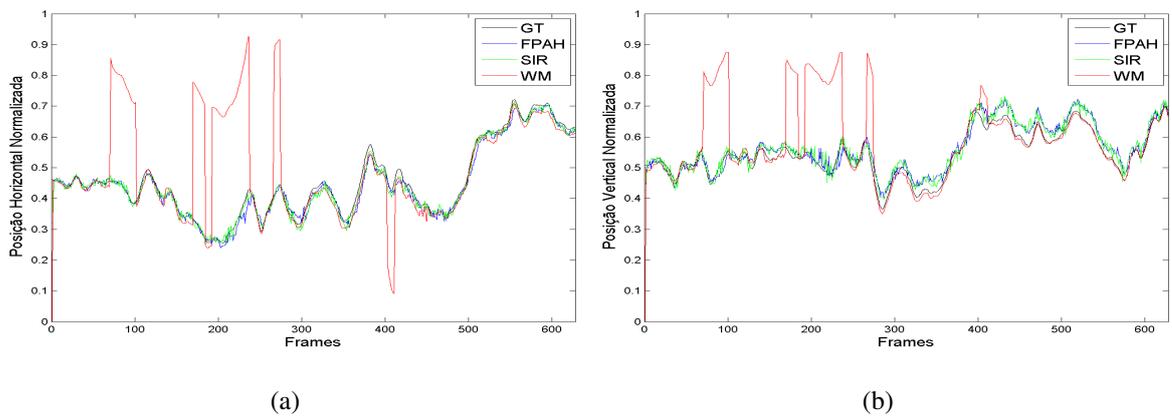


Figura 5.26: Sequência com variação de *background*. *Ground Truth* e estimaco da posio dos algoritmos WM, SIR e FPAH orientao Horizontal(a) e Vertical(b).

rastreamento com sucesso nos *frames* antes da oclusão. Entretanto no momento da oclusão o processo de estimação perde o alvo, fazendo com que o rastreamento fique comprometido. Em relação ao rastreamento realizado pelo filtro FPAH, este apresenta melhor desempenho conseguindo fazer a estimação com a utilização da informação inercial inserida nas partículas. Uma outra característica interessante da implementação do filtro FPAH é que, ao se fazer uso do algoritmo de Metropolis-Hastings, assegura-se que o movimento das partículas seja próximo à posição do objeto, reduzindo assim a utilização em demasia do algoritmo de *resampling*.

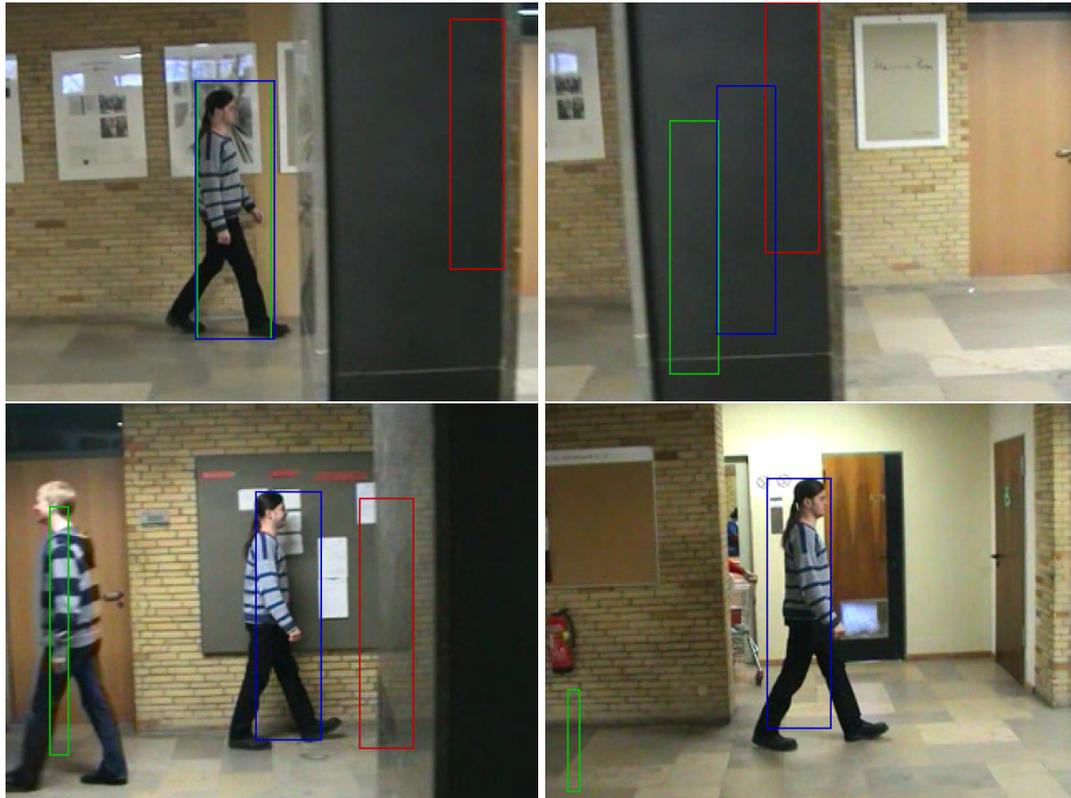


Figura 5.27: Sequência com oclusão total (*indoor*): WM, SIR e FPAH.

Nos gráficos que descrevem o movimento realizado na sequência (Figura 5.28) observa-se que o algoritmo WM, novamente apresenta dificuldades em realizar o rastreamento com sucesso. Embora o filtro SIR consiga realizar o rastreamento, sua estimação é muito distante do valor real da posição fornecida pelo *ground truth*. Já o filtro FPAH apresenta uma resposta bem próxima do valor real em relação ao *ground truth*, além de manter uma variação mais suave e consistente com o movimento do alvo na sequência.

Para reforçar os testes para a verificação da metodologia proposta com as demais, a sequência da Figura 5.29 apresenta uma pessoa caminhando ao ar livre (*outdoor*) em que ocorrem diversas situações de oclusões parciais e totais. As situações apresentadas nesta sequência em muito contribui para a validação da robustez da metodologia proposta, pois caracteriza uma situação real de aplicação que pode ser encontrada por um sistema envolvendo rastreamento visual.

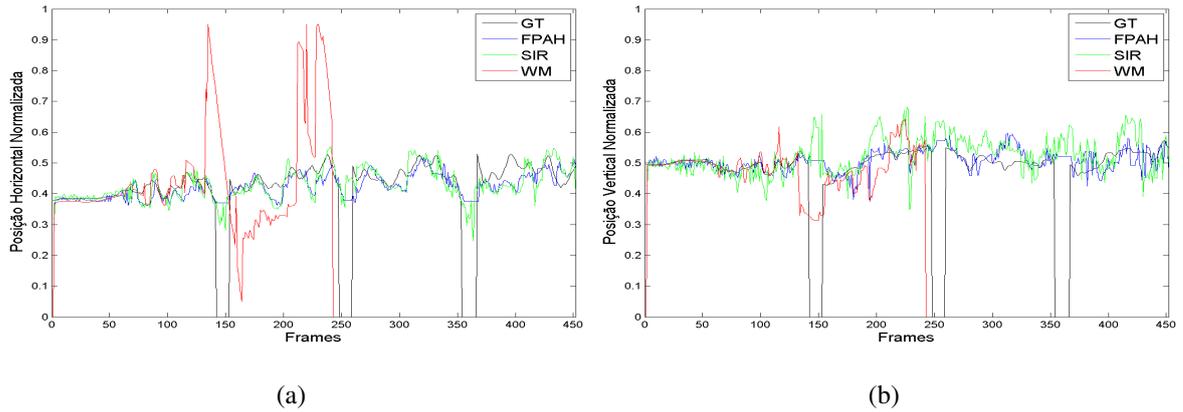


Figura 5.28: Sequência com oclusão total (*indoor*): *Ground Truth* e estimação da posição dos algoritmos *WM*, *SIR* e *FPAH* orientação Horizontal(a) e Vertical(b).

O algoritmo *WM* apresenta falha no rastreamento após a ocorrência de uma oclusão. Esta falha ocorre também na presença de pessoas com roupas similares as roupas da pessoa rastreada. O filtro *SIR* apresenta também algumas perdas do alvo depois das oclusões, bem como erros na estimação das dimensões do alvo no decorrer da sequência. O filtro *FPAH* mantém a estimação sobre a posição e as dimensões do alvo durante toda a sequência, realizando a partir do algoritmo de *reamostragem* as correções necessárias para o sucesso do rastreamento.

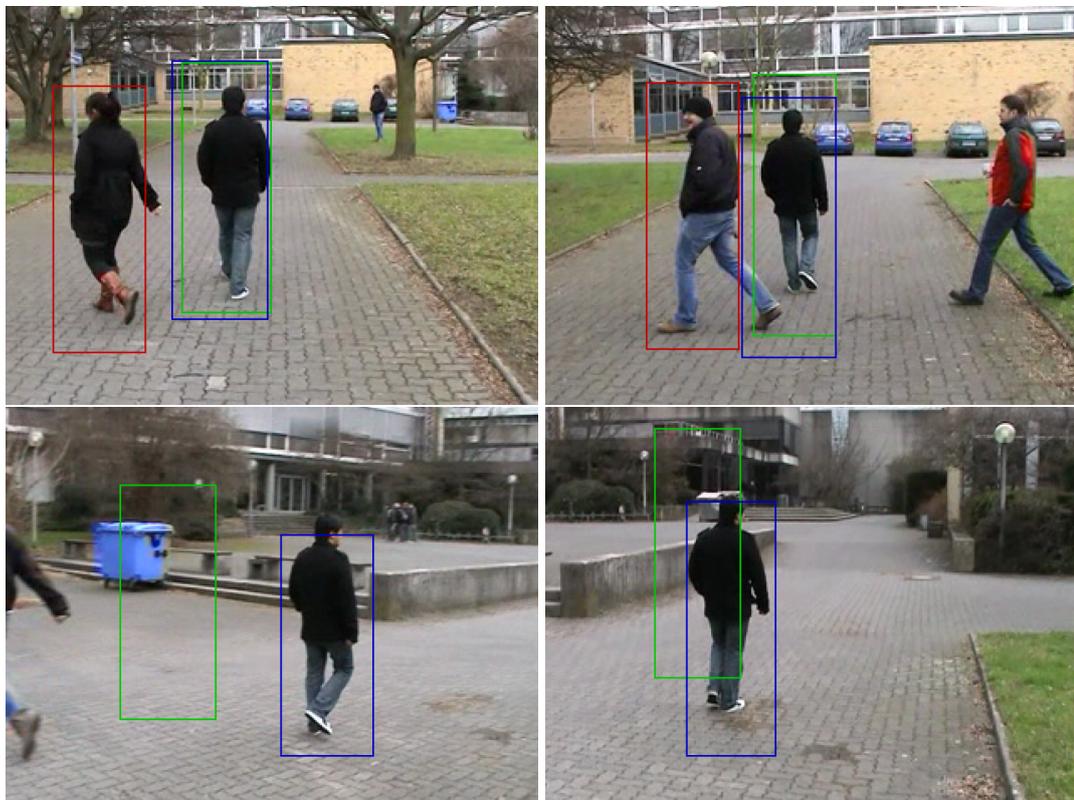


Figura 5.29: Sequência *outdoor* com oclusão total: *WM*, *SIR* e *FPAH*.

Nos gráficos obtidos dos resultados fornecidos pelos algoritmos avaliados (Figura 5.30), observa-se erros de rastreamento do algoritmo *WM*, depois de uma oclusão, em relação ao *Ground Truth*. O filtro *SIR* consegue realizar o rastreamento, mas com grandes variações das posições estimadas ao longo dos *frames*. O filtro *FPAH* realiza o rastreamento com sucesso, com pequenos valores de erros em relação ao *ground truth*. Após uma oclusão ocorrem variações das posições, mas devido à elevada robustez desta metodologia, o algoritmo consegue recuperar o alvo depois de poucos *frames*, realizando o rastreamento com sucesso.

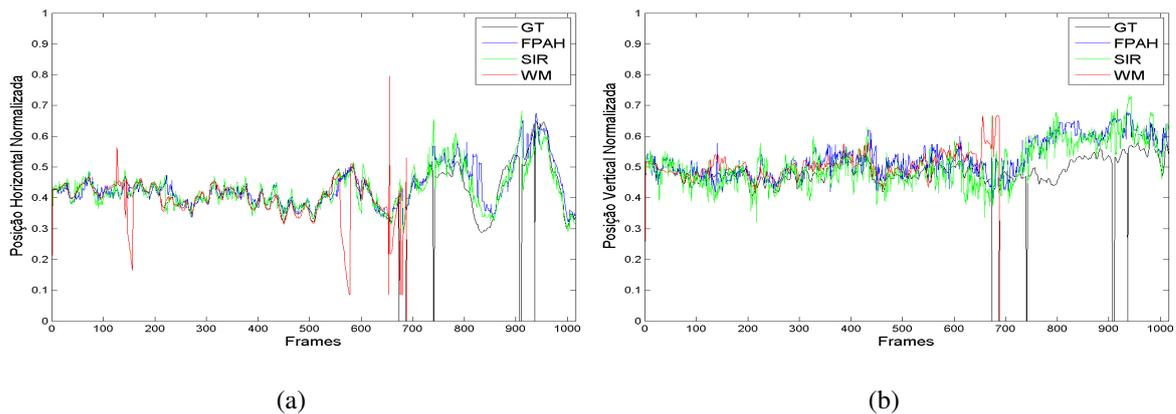


Figura 5.30: Sequência com oclusão *Ground Truth* e estimação da posição dos algoritmos *WM*, *SIR* e *FPAH* orientação Horizontal(a) e Vertical(b).

#### 5.8.5.1 Compilação dos resultados

Existem poucas metodologias que podem ser empregadas para a avaliação do desempenho de metodologias de rastreamento visual. Muitas delas são definidas para atender demandas específicas dificultando sua utilização em situações mais genéricas em que o rastreamento visual possa vir a ser aplicado. Desta feita, optou-se em utilizar como uma primeira abordagem comparativa o Erro Médio Quadrático e o Desvio Padrão da posição horizontal e vertical fornecida pelos algoritmos avaliados em relação ao *ground truth* de cada sequência utilizada, como descrito na Tabela 5.1.

Corroborando os resultados descritos nas Seções anteriores, a Tabela 5.1 apresenta valores menores de erro para o Filtro *FPAH* em quase em todas as sequências testadas, bem como o valor do desvio padrão obtido. Isto demonstra um melhor desempenho do filtro *FPAH* frente ao filtro *SIR* e ao *WM* na realização do rastreamento visual de objetos.

Seguindo a linha de avaliação dos resultados obtidos nas sequências testadas, utilizou-se também a avaliação da medida da variação da estimação da dimensão do alvo, avaliando a região definida pelo algoritmo que representaria o alvo. O *ground truth* utilizado (vide [48]) permite esta avaliação, pois além da posição espacial do alvo a ser rastreado, são fornecidas as dimensões espaciais da região de interesse que envolve o alvo. Este tipo de abordagem permite que seja avaliado o efeito de *drift* que ocorre em cada metodologia utilizada, sendo

	FPAH		SIR		WM	
	Erro Médio	$\sigma$	Erro Médio	$\sigma$	Erro Médio	$\sigma$
Sem Oclusão	<b>0,036</b>	<b>0,015</b>	0,045	0,023	0,043	0,115
Mud. <i>background</i>	<b>0,029</b>	<b>0,015</b>	0,033	0,020	0,1	0,18
Mud. Escala	0,042	0,029	<b>0,031</b>	<b>0,026</b>	0,159	0,1
Mud. Trajetória	<b>0,037</b>	<b>0,049</b>	0,082	0,111	0,063	0,126
Oclusão Total	<b>0,037</b>	<b>0,03</b>	0,048	0,037	0,13	0,11
Outdoor	<b>0,047</b>	<b>0,032</b>	0,051	0,039	0,086	0,09
Erro Médio Global	<b>0,036</b>	<b>0,028</b>	0,049	0,042	0,098	0,12

Tabela 5.1: Avaliação do erro médio quadrático e desvio padrão para os algoritmos avaliados em relação *ground truth*.

este parâmetro útil para definir a tendência de um algoritmo perder o alvo ou não. Quanto menor o efeito de *drift* mais robusto tende a ser a metodologia utilizada.

Com a intenção de medir esta característica foi utilizada a proposta apresentada em [50], que considera a relação entre a sobreposição da região de interesse fornecida pelos algoritmos testados em relação ao *Ground Truth*, como descrito pela Equação 5.1.

$$A(GT, ST) = \frac{Area(GT \cap ST)}{Area(GT \cup ST)} \quad (5.1)$$

Onde  $GT$  representa a área obtida para com os dados do *Ground Truth* e  $ST$  representa a área obtida para com os dados resultantes das metodologias de rastreamento visual avaliadas. De acordo com [50], considera-se como um verdadeiro positivo o valor obtido pela Equação 5.1 para  $A(GT, ST)$  acima de um valor de limiar  $T_{lim}$ . A escolha deste valor foi também definida por [50] como 20%. Na tabela 5.2 são apresentados os resultados percentuais de verdadeiros positivos obtidos em relação ao número total de *frames* onde esta presente o alvo, descartando os *que ocorrem oclusões*.

Um observação importante a ser feita nesta abordagem é que, devido ao modelo dinâmico de movimento utilizados pelos filtros SIR e FPAH permitirem a variação das dimensões espaciais da região de interesse, o que não ocorre no algoritmo WM, pode atrapalhar na avaliação por esta métrica. Entretanto, no caso dos testes realizados o algoritmo WM se demonstrou ineficiente em praticamente todas elas.

Pode-se constatar que na sequência do cubo o filtro SIR obteve maior porcentagem de verdadeiros positivos, mas a diferença em relação ao filtro FPAH é muito pequena em torno de 2%, o que pressupõe um desempenho similar nos resultados para esta sequência. E novamente, segundo a Tabela 5.2, observa-se que a porcentagem de verdadeiros positivos ao longo de cada sequência é maior para o filtro FPAH, corroborando o melhor desempenho e

	FPAH	SIR	WM
Sem Oclusão	<b>99,78</b>	95,97	93,65
Mud. <i>background</i>	<b>97,60</b>	95,21	82,93
Mud. Escala	92,30	<b>94,96</b>	33,90
Mud. Trajetória	<b>95,67</b>	76,87	87,18
Oclusão Total	<b>88,93</b>	86,06	30,08
Outdoor	<b>93,79</b>	90,18	70,07
Média Global	<b>94,72</b>	89,83	66,30

Tabela 5.2: Dados da porcentagem de verdadeiros positivos em relação ao número de *frames* considerando de sobreposição dos algoritmos avaliados e o *ground truth*.

robustez da metodologia proposta em relação às demais.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho abordou-se o problema do rastreamento visual de objetos móveis em uma sequência de imagens. Discutiram-se os principais métodos de rastreamento visual encontrados na literatura e optando-se pela implementação do método baseado em uma abordagem probabilística, especificamente envolvendo o método de Monte Carlo realizado pelo Filtro de Partículas.

Devido à natureza do rastreamento visual em imagens ser considerado uma tarefa de elevada complexidade, incluindo características não-lineares de obtenção do modelo apropriado, motivou a escolha pela utilização do Filtro de Partículas por se adequar a situações envolvendo não-linearidade e respostas adequadas para distribuições multimodais.

O método proposto e desenvolvido nesta dissertação emprega o modelo baseado no histograma normalizado de cor no espaço de cores *HSV*. A utilização do espaço de cores *HSV* foi realizada por apresentar um comportamento com menor sensibilidade em relação as mudanças de iluminação e melhor desempenho do rastreamento em ambientes externos sem iluminação controlada. A partir desta premissa, combinando características de metodologias específicas como a reamostragem limitada pelo algoritmo Metrópolis-Hassting, foi possível obter uma nova proposta o Filtro de Partículas de Amostragem Híbrida (FPAH). A principal característica desta abordagem se faz pela adaptação do algoritmo de Metrópolis-Hassting, aplicando a relação dos pesos como critério para aceitação e rejeição da nova amostragem. Com isto melhora-se a velocidade de resposta do algoritmo por evitar o excessivo emprego da reamostragem, fazendo com que as partículas permaneçam próximas da melhor estimativa a cada iteração.

Outra característica alcançada pela metodologia proposta se faz em relação à robustez, principalmente para o tratamento de oclusões totais e parciais. Nesta abordagem o algoritmo utiliza a informação referente ao movimento inercial das partículas, ponderando as diferenças do movimento de cada uma das partículas. Com a ajuda do algoritmo *Metrópolis*, assegura-se que o módulo e a direção da variação da posição, correspondam à melhor estimativa.

Para os testes realizados optou-se pela realização de dupla amostragem, inercial e de posição da melhor partícula, com o objetivo de abranger um maior número de possibilidades de movimentação do alvo, acrescentando além a velocidade de movimentação do conjunto de partículas para casos de movimentos com mudanças rápidas de posição. Outra melhoria alcançada em relação a robustez foi modelar o comportamento inercial similar à mudança rápida de posição, permitindo a correção da dimensão da região de interesse em situações em que ocorrem variação de escala do alvo na imagem durante o rastreamento.

Para avaliação do algoritmo foram utilizadas as sequências do *Benchmark BoBot* por

apresentar as condições que podem ocorrer em uma aplicação com situações reais de rastreamento visual, que abrange desde rotações, mudanças de iluminação, alterações rápidas de trajetórias, até a ocorrência de oclusões parciais e totais, todas estas com *ground truth*. A escolha destas seqüências aproxima a aplicação dos algoritmos desenvolvidos a situações reais, ou seja 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.

A metodologia proposta do filtro FPAH também oferece 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 independentemente da plataforma de desenvolvimento, permitindo ao desenvolvedor a escolha da linguagem e ambiente de desenvolvimento que melhor convier.

Tendo em vista o desempenho do algoritmo, bem como a partir de sua análise de complexidade, é facilmente plausível implementar um sistema de rastreamento em tempo real, tendo como fator limitador somente o número de partículas utilizada no filtro.

O trabalho realizado permitiu também verificar, que seria pretencioso chegar a uma solução definitiva para um algoritmo de rastreamento visual. Ainda não existe na literatura este tipo de algoritmo. Os algoritmos desenvolvidos aqui são de fácil implementação e configuração e podem ser aplicados em uma grande gama de aplicações, em que seja necessária a utilização de rastreamento de objetos em imagens.

Outro passo evidente para a continuação do trabalho é sua implementação em uma plataforma computacional dedicada, na qual permita a avaliação em situações reais, tais como navegação robótica. Para tal, faz-se necessária a implantação de uma arquitetura computacional especialista em controle de tempo real (ex.: Linux Xenomai).

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] A. Yilmaz and a. M. S. Omar Javed, “Object tracking: A survey,” *ACM Computing Surveys*, vol. 38, p. 45, 2006.
- [2] D. Gavrilu, “Pedestrian detection from a moving vehicle,” *ECCV*, 2000.
- [3] J. Fan, Y. Wu, and S. Dai, “Discriminative spatial attention for robust tracking,” *ECCV*, 2010.
- [4] J. Wang and Y. Yagi, “Visual tracking and segmentation using appearance and spatial information of patches,” *IEEE International Conference on Robotics and Automation*, 2010.
- [5] <http://tudosobreacor.blogspot.com.br/2011/02/modelo-hsv-caracterizacao.html>.
- [6] G. Wang, D. Xiao, and J. Gu, “Review on vehicle detection based on video for traffic surveillance,” *IEEE Automation and Logistics*, pp. 2961 – 2966, 2008.
- [7] X. Gao, D. You, and S. Katayama, “Seam tracking monitoring based on adaptive kalman filter embedded elman neural network during high-power fiber laser welding,” *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, vol. 59, pp. 4315–4325, 2012.
- [8] C.-Y. Chang and H. W. Lie, “Real-time visual tracking and measurement to control fast dynamics of overhead cranes,” *IEEE INDUSTRIAL ELECTRONICS*, vol. 59, pp. 1640–1649, 2012.
- [9] X. Xiang, “A brief review on visual tracking methods,” *Intelligent Visual Surveillance*, pp. 41 – 44, 2011.
- [10] W. Ge, R. T. Collins, and R. B. Ruback, “Vision-based analysis of small groups in pedestrian crowds,” *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 34, pp. 1003–1016, 2012.
- [11] B. Leibe, K. Schindler, N. Cornelis, and L. V. Gool, “Coupled object detection and tracking from static cameras and moving vehicles,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1683 – 1698, 2012.
- [12] Y. Kurylyak, F. Lamonaca, and G. Mirabelli, “Detection of the eye blinks for human’s fatigue monitoring,” *IEEE International Symposium Medical Measurements and Applications Proceedings*, vol. 1-4, pp. 2102–2107, 2012.

- [13] D. Stein, H. Monnich, J. Raczkowski, and H. Worn<sup>1</sup>, “Visual servoing with an optical tracking system and a lightweight robot for laser osteotomy,” *IEEE International Conference on Control and Automation*, pp. 1896–1900, 2009.
- [14] W. G. Pence, F. Farelo, R. Alqasemi, Y. Sun, and R. Dubey, “Visual servoing control of a 9-dof wrra to perform adl tasks,” *IEEE International Conference on Robotics and Automation*, pp. 916 – 922, 2012.
- [15] F. Bonin-Font, A. Burguera, A. Ortiz, and G. Oliver, “Combining obstacle avoidance with robocentric localization in a reactive visual navigation task,” *IEEE Industrial Technology*, pp. 19 – 24, 2012.
- [16] S. Bousnina, B. Ammar, N. Baklouti, and A. M. Alimi, “Learning system for mobile robot detection and tracking,” *The 2nd International Conference on Communications and Information Technology*, pp. 384 – 389, 2012.
- [17] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, “Recent advances and trends in visual tracking: A review,” *Neurocomputing*, vol. 74, 2011.
- [18] 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.
- [19] T. Ojala, M. P. inen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, pp. 972–987, 2002.
- [20] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, “Evaluation of local spatio-temporal features for action recognition,” *ICCV*, 2009.
- [21] A. Klaser, M. Marszalek, and C. Schmid, “A spatio-temporal descriptor based on 3d-gradients,” *BMVC*, 2008.
- [22] O. Tuzel, F. Porikli, and P. Meer, “Human detection via classification on riemannian manifolds,” *CVPR*, 2007.
- [23] L. Andersson and R. Howard, “Comparison and rigidity theorems in semi-riemannian geometry,” *Comm. Anal. Geom.*, vol. 6, pp. 819–877, 1996.
- [24] Y. Huang, K. Huang, L. Wang, D. Tao, T. Tan, and X. Li, “Enhanced biologically inspired model,” *CVPR*, 2008.
- [25] F. B. Vidal and V. H. C. Alcalde, “Window-matching techniques with Kalman filtering for an improved object visual tracking,” in *Third annual IEEE Conf. on Automation Science and Eng., IEEE CASE 2007*, vol. 22-25, September 2007, pp. 829–834.

- [26] A. Giachetti, "Matching techniques to compute image motion," *Image and Vision Computing*, vol. 18, no. 3, pp. 247–260, February 2000.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] A. Zylberberg, *Probabilidad y Estadística*. Nueva Libreria, 2005.
- [34] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, Ed., 2006.
- [35] D. A. Forsyth and J. Ponce, *Computer Vision: A modern approach*. Prentice-Hall, 2003.
- [36] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal procesing*, vol. 50, 2002.
- [37] S. Imtiaz, K. Roy, B. Huang, S. Shah, and P. Jampana, "Estimation of states of nonlinear systems using a particle filter," in *Industrial Technology, 2006. ICIT 2006. IEEE International Conference on*, dec. 2006, pp. 2432 –2437.
- [38] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo methods for bayesian filtering," *Statistics and Computing*, vol. 10, p. 197–208, 2000.
- [39] J. D. Hol, T. B. Schon, and F. Gustafsson, "On resampling algorithms for particle filters," in *Nonlinear Statistical Signal Processing Workshop*, 2006.

- [40] G. Kitagawa, “Monte carlo filter and smoother for non-gaussian nonlinear state space models,” *Journal of Computational and Graphical Statistics*, vol. 5, pp. 1–25, 1996.
- [41] S. Chib and E. Greenberg, “Understanding the Metropolis-Hastings Algorithm,” *The American Statistician*, vol. 49, no. 4, pp. 327–335, 1995. [Online]. Available: <http://dx.doi.org/10.2307/2684568>
- [42] O. Straka and M. Šimandl, *Using the Bhattacharyya distance in functional sampling density of particle filter*. Oxford: Elsevier, 2006, vol. 1, pp. 1006–1011. [Online]. Available: [http://www.kky.zcu.cz/en/publications/StrakaO\\_2006\\_Usingthe](http://www.kky.zcu.cz/en/publications/StrakaO_2006_Usingthe)
- [43] N. A. Thacker, F. J. Aherne, and P. I. Rockett, “The bhattacharyya metric as an absolute similarity measure for frequency coded data.” *Kybernetika*, vol. 34, pp. 363–368, 1997.
- [44] C. H. P. Perez, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking,” *ECCV*, pp. 661–675, 2002.
- [45] J. Li and C.-S. Chua, “Transductive inference for color-based particle filter tracking,” *Image and Vision Computing*, vol. 3, pp. 949–952, 2003.
- [46] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Algoritmos Teoria e Prática*. Elsevier Editora Ltda, 2002.
- [47] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.
- [48] [Online]. Available: <http://www.iai.uni-bonn.de/~kleind/tracking/>
- [49] G. Bradski and A. Kaehler, *Learning OpenCV Computer Vision with the OpenCV Library*, M. Loukides, Ed. O’Reilly, 2008.
- [50] F. Yin, D. Makris, and S. Velastin, “Performance evaluation of object tracking algorithms,” *Proceeding Tenth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2007.

## APÊNDICES

# A. ANÁLISE DA COMPLEXIDADE COMPUTACIONAL

## A.1 METODOLOGIA EMPREGADA

Para se realizar as análises de complexidades algumas aproximações são consideradas: quantidade de memória física disponível na arquitetura computacional utilizada, sequencialidade e se não existem execuções simultâneas ou em paralelo.

O tempo de execução de uma entrada é o tempo empregado na execução de cada uma das etapas. Cada linha de código requer um tempo  $c$  para ser executada. Para facilitar o análise, este tempo é igual e independente da linha de execução. Para casos nos quais existem laços condicionais para a execução, escolherá-se-á o pior deles, por considerar que este é o tempo máximo que o algoritmo vai empregar. Ainda considera-se que o pior dos casos ocorre com elevada freqüência de repetição. Quando existem *loops* aninhados, a ordem de complexidade aumenta em proporção aos ciclos concatenados no expoente que acompanha ao numero de entrada.

## A.2 ANALISE DO FILTRO SIR(SAMPLING IMPORTANCE RESAMPLING)

1. Instrução	Custo	Vezes
2. <b>Para</b> $i = 1 : N_s$	$C_1$	$n$
3. $X_{i:k} \sim p(X_k   X_{i:k-1})$	$C_2$	$n - 1$
4. $w_i^* = P(Y_k   X_{i:k})$	$C_3$	$n - 1$
5. <b>Para</b> $i = 1 : N_s$	$C_4$	$n$
6. $t = \sum_i^{N_s} w_{i:k}$	$C_5$	$n - 1$
7. $N = \sum_i^{N_s} w_{i:k}^2$	$C_6$	$n - 1$
8. $N_{eff} = \frac{1}{N}$	$C_7$	1
9. <b>Re-amostragem</b>	$C_8$	$n$
10. <b>Para</b> $i = 1 : N_s$	$C_9$	$n$
11. $\hat{X}_k = \sum_i^{N_s} w_{i:k} X_{i:k}$	$C_{10}$	$n - 1$

De acordo com os resultados do tempo de execução para cada linha do algoritmo, obtém-

se um resultado condensado que contem coeficientes representando os valores das linhas, em que executam entradas com igual ordem do expoente. Então o custo computacional ficará definido da seguinte maneira:

$$C_c = An + B \quad (\text{A.1})$$

Para obter um resultado mais representativo, utiliza-se a maior ordem do polinômio, considerando que é o valor que mais contribui, em proporção, com o tamanho da entrada. Este valor corresponde a 1, então a complexidade computacional para o SIR fica igual a  $O(n)$ , onde  $O$  é um coeficiente que não depende do tamanho da entrada e representa o limite superior no tempo de execução do algoritmo.

### A.3 ANÁLISE REAMOSTRAGEM

1. <b>Instrução</b>	<b>Custo</b>	<b>Vezez</b>
2. <b>Para</b> $i = 2 : N_s$	$C_1$	$n - 1$
3. $c_i = c_{i-1} + w_{i:k}$	$C_2$	$n - 2$
4. <b>Para</b> $j = 1 : N_s$	$C_3$	$n$
5. $u_j = u_1 + N_s^{-1}(j - 1)$	$C_4$	$n - 1$
6. <b>While</b> $u_j > c_i$	$C_5$	$n - 1$
7. $i = i + 1$	$C_6$	$n - 1$
8. $X_j = X_i$	$C_7$	$n - 1$

Assume-se que as linhas de execução empregam o mesmo tempo. Na linha 6, tem-se um laço condicional e para analisá-lo, considera-se o pior dos casos no qual  $u_j$  é sempre maior do que  $c_i$ . No final do filtro, isto significa que a primeira partícula é a de maior peso atribuído. Tendo em conta isso para o Reamostragem, o valor do tempo de execução é dado como complexidade  $O(n)$ .