

**UNIVERSIDADE FEDERAL DE PELOTAS**

Bacharelado em Ciência da Computação



**Trabalho Acadêmico**

**Visão Computacional: Estudo do Filtro de Kalman em  
Conjunto ao Mean Shift no Rastreamento de Jogadores  
em Esportes Coletivos.**

**Marcelo Horne Mattos**

MARCELO HORNE MATTOS

**VISÃO COMPUTACIONAL: ESTUDO DO FILTRO DE KALMAN EM CONJUNTO  
AO MEAN SHIFT NO RASTREAMENTO DE JOGADORES EM ESPORTES  
COLETIVOS.**

Trabalho acadêmico apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

PELOTAS  
2007

**Dados de catalogação na fonte:**


Ubirajara Buddin Cruz – CRB-10/901  
Biblioteca de Ciência & Tecnologia - UFPel

M444v      Mattos, Marcelo Horne

Visão computacional : estudo do filtro de Kalman em conjunto ao Mean Shift no rastreamento de jogadores em esportes coletivos / Marcelo Horne Mattos ; orientador Anderson Ferrugem ; co-orientador Lucas Ferrari de Oliveira. – Pelotas, 2007. – 88f. - Monografia (Conclusão de curso). Curso de Bacharelado em Ciência da Computação. Departamento de Informática. Instituto de Física e Matemática. Universidade Federal de Pelotas. Pelotas, 2007.

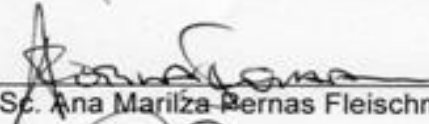
1.Informática. 2.Visão computacional. 3. Rastreamento. 4.Oclusão. 5.Sistema Kinesis 6.Mean Shift. 7.Filtro de Kalman I.Ferrugem, Lucas. II.Oliveira, Lucas Ferrari de. III.Título.

**BANCA EXAMINADORA:**



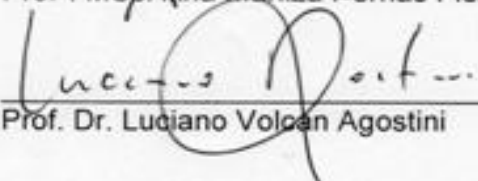
---

Prof. MSc. Anderson Priebe Ferrugem



---

Profª. MSc. Ana Marilza Bernas Fleischmann



---

Prof. Dr. Luciano Volcan Agostini

## **AGRADECIMENTOS**

Primeiramente, e principalmente, aos meus pais, que nunca mediram esforços pra me proporcionar tudo o que eu precisasse para construir uma carreira, uma vida. Espero sinceramente um dia ser capaz de retribuir e honrar todo este sacrifício, e oferecê-lo para meus filhos da mesma forma que me foi oferecido.

Agradeço à minha irmã, pela tradicional “admiração de irmão mais novo”, que apesar de nem sempre eu demonstrar, me enche de orgulho e me dá forças para continuar a ser um bom exemplo.

Aos demais membros da minha família, que sempre torceram por mim ao longo de toda esta jornada, e de outras que virão.

Aos meus amigos, sem os quais eu não teria motivos pra sorrir, nem alegria suficiente para iniciar mais um dia de aulas ou de diversão. Em especial aos meus colegas de turma do início ao fim: Bruno, Cléber, Eduardo, Giulian, Helen, Juliano, Leandro, Lucas, Mateus, Matheus, Rodrigo e Vinícius. Também ao meu colega de orientação e formatura Gustavo.

Ao meu orientador, Prof. Anderson Priebe Ferrugem, pela ajuda indispensável e constante, além da amizade demonstrada nos momentos de dificuldade, que não foram poucos.

Ao meu Prof. Co-orientador Lucas Ferrari de Oliveira, também pela ajuda fornecida e pelos demais conhecimentos passados como professor ao longo do curso.

À Prof<sup>a</sup> Flávia Azambuja, pela ajuda com os estágios ao longo do curso, além da atenção dispensada sempre que foi solicitada.

A todos os outros professores, indispensáveis para que eu atingisse um nível de conhecimento suficiente para concluir o ensino superior.

E, ao contrário do que se faz de costume, agradeço por último a Deus, por ter dado o dom da vida a todos os que eu já agradei. Agradeço a Ti só agora, porque todos os outros agradecimentos também são a Ti, indiretamente

*Dedico este trabalho à memória de minha avó Odette, que sempre disse que me queria ver crescido, formado, sendo alguém na vida e, sobretudo, uma pessoa boa. Espero que do lugar melhor onde ela esteja, possa sentir orgulho de mim e realizado esta vontade. És vó, sem dúvida, a pessoa que mais faz falta ao meu lado.*

## Resumo

MATTOS, Marcelo Horne. **Visão Computacional: Estudo do Filtro de Kalman em Conjunto ao Mean Shift no Rastreamento de Jogadores em Esportes Coletivos.** 2007. 88f. Monografia (Bacharelado em Ciência da Computação). Universidade Federal de Pelotas, Pelotas.

Sistemas computacionais têm sido constantemente requisitados na área esportiva atual. Ferramentas capazes de avaliar um atleta durante uma partida, fornecendo dados estatísticos a respeito do seu desempenho, são de grande importância para treinadores e preparadores físicos. O sistema Kinesis tem o objetivo de ser uma destas ferramentas, trabalhando a partir de vídeos de jogos de esportes coletivos, efetuando um rastreamento baseado no algoritmo de *mean shift* (MS), que é uma técnica para achar modas em um conjunto de amostras de dados reais, manifestando uma função de densidade de probabilidade. Porém o MS apresenta algumas falhas de execução, em especial em situações de oclusão e/ou deformação do alvo do rastreamento. O filtro de Kalman (FK) objetiva solucionar estes problemas de forma eficiente, partindo de uma primeira estimativa baseada na posição inicial do alvo, de forma a prever as posições futuras. Conceitos de visão computacional e rastreamento foram estudados, bem como a versão inicial do sistema Kinesis e o algoritmo de *mean shift*. Além disso, foi dado foco ao estudo detalhado do filtro de Kalman padrão e suas extensões não-lineares, visando à inclusão desta técnica em conjunto com o MS. Devido às características particulares do MS e do FK, propõe-se um algoritmo para união de ambas as técnicas, com o intuito de que uma possa suprir as deficiências da outra de forma robusta, melhorando o desempenho do sistema Kinesis. O algoritmo foi descrito passo a passo, separado em seções, e apresenta de forma clara cada ação e cuidado necessários para uma futura implementação, idéia fortalecida por testes parciais realizados paralelamente a este trabalho.

Palavras-chave: Visão Computacional. Rastreamento. Oclusão. Sistema Kinesis. *Mean Shift*. Filtro de Kalman.

## **Abstract**

MATTOS, Marcelo Horne. **Visão Computacional: Estudo do Filtro de Kalman em Conjunto ao Mean Shift no Rastreamento de Jogadores em Esportes Coletivos.** 2007. 88f. Monografia (Bacharelado em Ciência da Computação). Universidade Federal de Pelotas, Pelotas.

Computing Systems have been constantly required in the actual sportive field. Tools that are able to evaluate an athlete during a match and giving statistic data about his performance are very important to trainers and physical preparators. The Kinesis system has the goal of being one of these tools, working from collective sports matches' videos, performing a tracking based on the mean shift (MS) algorithm, which is a technique to find modes in a set of real data samples, manifesting a probability density function. However the MS presents some execution errors, especially in cases of occlusion or shape distortion of the tracking target. The Kalman filter (KF) objectives to solve these problems efficiently, starting from a first estimation based on the initial position of the target to estimate future positions. Computer Vision and tracking concepts had been studied, the same way as the first version of the Kinesis system and the mean shift algorithm. Furthermore a focus had been set on a study of the standard Kalman filter and its nonlinear extensions, aiming a combination of this technique with the mean shift algorithm. Based on the particular characteristics of the MS and FK was proposed an algorithm to adjoin both techniques in order to one helping another against their deficiencies in a robust way, improving the performance of the Kinesis system. The algorithm was described step by step, separated in sections, presenting clearly each action and caution necessities to a future implementation, reinforced by parallel partial tests.

Keywords: Computer Vision. Tracking. Occlusion. The Kinesis System. Mean Shift. Kalman Filter.



## Lista de Figuras

Figura 1 – Síntese dos Trabalhos Relativos aos Métodos Baseados em Estimativa Visual .....	26
Figura 2 – Síntese dos Trabalhos Relativos aos Métodos Baseados em Sensores-Transmissores.....	27
Figura 3 – Síntese dos Trabalhos Relativos aos Métodos Baseados em Imagens ...	28
Figura 4 – Região de interesse, centro de massa e vetor <i>mean shift</i> .....	30
Figura 5 – Convergência no procedimento de <i>mean shift</i> .....	31
Figura 6 – Extração de dados de um conjunto real .....	32
Figura 7 – Estimação da PDF.....	32
Figura 8 – Gráfico Perfil de Epanechnikov .....	34
Figura 9 – Gráfico do coeficiente de Bhattacharyya para teste de oclusão total .....	37
Figura 10 – Sistema de Cores RGB .....	40
Figura 11 – Coordenadas do sistema HSB, com saturação 0,5.....	41
Figura 12 – Estimação de estado: extração de informação e aprimoramento.....	49
Figura 13 – Ciclo de operação do filtro de Kalman discreto .....	50
Figura 14 – Estrutura detalhada do ciclo do filtro de Kalman .....	51
Figura 15 – Ciclo de operação do filtro de Kalman discreto .....	52
Figura 16 – Conceitos Fundamentais para o filtro de Kalman.....	54
Figura 17 – Oclusão quase total do alvo .....	67
Figura 18 – Oclusão total do alvo.....	68
Figura 19 – Fluxograma do algoritmo de união do FK e MS .....	70
Figura 20 – Simulação do Kinesis na ferramenta MATLAB – Tela de seleção do alvo .....	73
Figura 21 – Simulação do Kinesis na ferramenta MATLAB – Tela de seleção do fundo .....	74

Figura 22 – Histograma do modelo do alvo gerado pelo sistema Kinesis .....	75
Figura 23 – Gráfico da velocidade do alvo rastreado ao longo dos <i>frames</i> .....	78

## **Lista de Tabelas**

Tabela 1 – Módulos funcionais da visão humana e de máquina .....	20
Tabela 2 – Categorias de movimento a partir da velocidade .....	45
Tabela 3 – Expansão em série de Taylor de $\cos(x)$ para valores variáveis de $x$ .....	61

## **Lista de Abreviaturas e Siglas**

2D – Duas Dimensões

3D – Três Dimensões

CMYK – *Cyan, Magenta, Yellow and Black*

FK – Filtro de Kalman

FKE – Filtro de Kalman Estendido

GPS – *Global Positional Systems*

HLS – *Hue, Lightness and Saturation*

HSB – *Hue, Saturation and Brightness*

HSV – *Hue, Saturation and Value*

MS – *Mean Shift*

PDF – *Probability Density Function*

RGB – *Red, Green and Blue*

UFPEL – Universidade Federal de Pelotas

UFRGS – Universidade Federal do Rio Grande do Sul

## Sumário

RESUMO.....	4
ABSTRACT .....	5
LISTA DE FIGURAS .....	6
LISTA DE TABELAS .....	8
LISTA DE ABREVIATURAS E SIGLAS .....	9
1 Introdução .....	15
1.1 Motivação.....	16
1.2 Objetivos .....	17
1.2.1 Objetivos Específicos .....	17
1.3 Organização do Trabalho .....	17
2 Visão Computacional .....	19
2.1 Definição .....	20
2.2 Componentes de um Sistema de Visão .....	21
2.3 Aplicações .....	22
3 Rastreamento.....	24
3.1 Definição .....	24
3.2 Aplicações .....	25
3.3 Classes de Métodos de Rastreamento .....	26
3.3 Revisão Bibliográfica sobre Sistemas de Rastreamento.....	27
4 Kinesis e <i>Mean Shift</i> .....	31
4.1 Mean Shift .....	31
4.2 Função Densidade de Probabilidade .....	33
4.2.1 Estimativa de Densidade de <i>Kernel</i> .....	34
4.2.2 Estimativa do Gradiente da Densidade .....	36
4.3 Métrica Baseada no Coeficiente de Bhattacharyya.....	38
4.4 Representação do Alvo .....	39
4.4.1 Modelo do Alvo.....	39

4.4.2 Candidato do Alvo .....	40
4.5 Representação de Cores.....	41
4.5.1 Sistema RGB.....	41
4.5.2 Sistema HSB (ou HSV) .....	42
4.5.3 Conversão RGB para HSB.....	44
4.6 Representação do Fundo.....	44
4.7 Minimização da Distância.....	45
4.8 Extração de Métricas.....	46
4.9 Algoritmo .....	48
4.10 Situações Críticas .....	49
5 O Filtro de Kalman .....	50
5.1 O Processo de Estimação Utilizando Filtros.....	50
5.2 Ciclo de Operação e Recursividade .....	52
5.3 Sistemas Lineares e o Filtro de Kalman Padrão .....	54
5.3.1 Exemplo Intuitivo .....	56
5.3.2 Sistemas Estáticos .....	58
5.3.3 Sistemas Dinâmicos .....	59
5.4 Sistemas Não-Lineares .....	61
5.4.1 Linearização pela Expansão em Série de Taylor .....	62
5.4.2 O Filtro de Kalman Linearizado .....	64
5.4.3 O Filtro de Kalman Estendido.....	66
6 O Filtro de Kalman como Apoio ao <i>Mean Shift</i> .....	68
6.1 Problemas de Oclusão e Deformação do Alvo.....	68
6.2 Motivos para Inclusão do Filtro de Kalman.....	70
6.3 Algoritmo para Adaptação do Filtro de Kalman ao Kinesis .....	71
6.3.1 Ajuste Inicial das Variáveis Globais do Sistema.....	73
6.3.2 Ajuste Inicial das Variáveis do <i>Mean Shift</i> .....	73
6.3.3 Ajuste Inicial das Variáveis do Filtro de Kalman.....	73
6.3.4 Seleção do Alvo .....	75
6.3.5 Seleção do Fundo .....	75
6.3.6 Geração do Perfil Epanechnikov do Modelo do Alvo .....	76
6.3.7 Conversão de Cores.....	76
6.3.8 Geração do Histograma do Modelo do Alvo.....	77
6.3.9 Traçado do Retângulo em Torno do Alvo.....	77
6.3.10 Visualização do <i>Frame</i> Atual.....	77
6.3.11 Estimativa Inicial do Filtro de Kalman.....	78
6.3.12 Laço de Iteração.....	78

6.3.13 Próxima Estimativa do Filtro de Kalman.....	78
6.3.14 Geração do Histograma do Candidato do Alvo .....	78
6.3.15 Comparação de Histogramas pelo Coeficiente de Bhattacharyya .....	79
6.3.16 Chamando o Procedimento de <i>Mean Shift</i> .....	79
6.3.17 Determinando o Novo Centro .....	79
6.3.18 Passos Finais .....	80
6.3.19 Geração de Gráficos .....	80
7 Conclusão .....	81
7.1 Trabalhos Futuros .....	84

## 1 Introdução

Recursos computacionais são cada vez mais requisitados na área esportiva. Isto se deve à necessidade de se ter mecanismos auxiliares aos mecanismos de testes físicos e de desempenho mais comuns, como suporte aos treinadores e preparadores físicos. Em geral, a avaliação de atletas de esportes coletivos é feita de forma isolada em testes físicos e de desempenho utilizando mecanismos *ad hoc* de avaliação. Além disso, sabe-se que existe uma grande diferença entre o desempenho de um atleta nos treinos e seu desempenho nos jogos, por fatores psicológicos ou mesmo técnicos.

Sendo assim, o sistema Kinesis teve seu desenvolvimento iniciado no Laboratório de Robótica Inteligente da Universidade Federal do Rio Grande do Sul (UFRGS), sob orientação do Prof. Dante Barone, com o objetivo final de ser um sistema em tempo real de rastreamento de jogadores em esportes coletivos, a partir de vídeos de jogos destes esportes. Em sua etapa inicial o Kinesis obteve um resultado satisfatório, porém alguns problemas no rastreamento foram encontrados durante sua fase de testes. Em suma, estes problemas ocorreram nas situações em que o alvo do rastreamento era ocluso ou sofria alteração em sua forma.

“O *mean shift* é um algoritmo de rastreamento de imagens usado para achar modas em conjunto de dados, manifestando uma função probabilística de densidade” (COMANICIU; RAMESH; MEER, 2000). No sistema Kinesis, o *mean shift* é a base do rastreamento por tratar-se de um processo vantajoso, pois não é necessária a avaliação da imagem por inteiro, ganhando, portanto, em rapidez. Esta técnica é apresentada com mais detalhes no capítulo 4 deste trabalho, juntamente com o sistema Kinesis.

Segundo Brown e Hwang (1997), o filtro de Kalman foi criado por Rudolph E. Kalman em 1960, com o objetivo inicial de fornecer uma solução recursiva para



filtragem linear de dados discretos. Para isto, utiliza equações matemáticas que implementam um estimador preditivo de estados, buscando corrigir iterativamente a resposta de um determinado sistema através de múltiplas variáveis relacionadas a ele. Suas áreas de aplicação são muito diversificadas, tais como: processamento de imagem, supervisores de eventos discretos, processamento de sinais, rastreamento, sistemas de inferência, etc.

Desta forma, este trabalho propõe o estudo do filtro de Kalman, como apoio ao algoritmo de *mean shift*. Com base nisto, um algoritmo para união destas duas técnicas é proposto, em busca de uma melhora significativa no desempenho do sistema em termos de qualidade de rastreamento.

## 1.1 Motivação

Esportes, em geral, movem multidões mundo afora. Tratando-se de esportes coletivos, sabe-se que existem muitas pessoas envolvidas neste trabalho, em uma incessante busca de resultados positivos.

Um fator extremamente relevante neste processo é a avaliação dos atletas, por parte de seus treinadores ou preparadores físicos. Para isto, são utilizados diversos mecanismos tradicionais no meio esportivo, porém sistemas computacionais capazes de auxiliar nesta avaliação têm sido cada vez mais usados e requisitados.

Sabe-se, também, que a realidade do esporte no Brasil, embora promissora, é carente de recursos, o que torna de extrema importância a possibilidade de desenvolver um sistema capaz de avaliar os atletas que se adapte a esta carência, sem afetar seu desempenho. Em meio a diversos sistemas computacionais aplicados na área esportiva ao longo dos anos, foi iniciado o desenvolvimento do Kinesis, tratando-se de um sistema de rastreamento a partir de vídeos de jogadores em esportes coletivos, de forma a extrair métricas para uso posterior e tendo como uma de suas características o seu baixo custo computacional.

O desenvolvimento inicial deste projeto apresentou algumas falhas, de forma a afetar significativamente o nível de rastreamento ideal. Desta forma, a aplicação de novas técnicas ao sistema visa solucionar estes problemas. Sendo assim, o foco deste trabalho é o estudo do filtro de Kalman como apoio ao *mean shift* visando,

futuramente, oferecer aos profissionais da área esportiva uma ferramenta útil e acessível financeiramente, de forma a atingir um nível satisfatório de avaliação dos atletas, podendo potencializar o rendimento dos mesmos em eventos esportivos profissionais.

## 1.2 Objetivos

Este trabalho tem como objetivo principal o estudo da técnica do filtro de Kalman adaptável ao sistema inicial do Kinesis, de forma a solucionar alguns dos problemas encontrados durante os testes da fase inicial de desenvolvimento do sistema. Com base neste estudo, propõe-se um algoritmo para união do filtro de Kalman com o *mean shift*.

### 1.2.1 Objetivos Específicos

- a) Estudo de conceitos de visão computacional e rastreamento;
- b) Estudo do projeto inicial do sistema Kinesis;
- c) Estudo sobre o algoritmo de *mean shift*;
- d) Estudo da ferramenta MATLAB;
- e) Estudo sobre as técnicas de filtro de Kalman;
- f) Desenvolvimento de algoritmos de simulação da técnica de filtro de Kalman;
- g) Proposta de um algoritmo de adaptação do filtro de Kalman no Kinesis, em conjunto ao *mean shift*.

## 1.3 Organização do Trabalho

Dividido em sete capítulos, este trabalho apresenta as técnicas e conceitos estudados durante seu desenvolvimento e, com base nelas, propõe um algoritmo para adaptação do filtro de Kalman ao *mean shift*.

No capítulo 2, é apresentada uma introdução sobre a área de Visão Computacional, trazendo um breve conceito, suas aplicações, recursos e áreas afins.

Logo a seguir, no capítulo 3, discutem-se algumas técnicas de rastreamento, e faz-se uma revisão bibliográfica contendo alguns exemplos de sistemas de rastreamento desenvolvidos para a área esportiva ao longo dos anos.

No capítulo 4, é apresentado o sistema Kinesis, englobando suas características e objetivos. Nesta abordagem, é dada ênfase ao algoritmo de *mean shift*, utilizado como base para o rastreamento.

Posteriormente, no capítulo 5, é descrita a técnica de aplicação do filtro de Kalman, com foco especial na sua adaptação ao sistema Kinesis visando solucionar alguns dos problemas encontrados na primeira versão do projeto.

A seguir, no capítulo 6, os problemas de oclusão e deformação do alvo encontrados nos testes iniciais do sistema Kinesis são relatados. Além disso, motivos para a inclusão do filtro de Kalman como solução a estes problemas são citados, e um algoritmo passo a passo para implementação do FK em conjunto ao *mean shift* é proposto.

Por fim, no capítulo 7 são apresentadas as conclusões acerca deste trabalho, além de apontar contribuições e aprimoramentos futuros do sistema projetado.

## 2 Visão Computacional

Segundo Sebe et al. (2005), em um resumo histórico, tudo começou com processamento de imagens (*image processing*), na década de 60. Este tipo de processamento foi inspirado na execução de processamento de sinais e permaneceu bastante orientado à programação.

Na década de 70, o que se chamou de análise de imagens (*image analysing*) surgiu combinando a medição de imagens com o reconhecimento de padrões baseado em estatísticas. Lentamente, os métodos computacionais iam separando-se dos sensores com o objetivo de se tornarem adequados a uma maior quantidade de necessidades.

Nos anos 80, a visão computacional dirigida ao modelo foi originada quando a inteligência artificial e a modelagem geométrica surgiram em conjunto com componentes de análise de imagens. A ênfase se deu em análises precisas, com pouca ou nenhuma interação, cujo valor era estimado principalmente pelos recursos visuais.

No início dos anos 90, a visão computacional ficou disponível a muitas pessoas graças ao advento de computadores suficientemente rápidos. A internet revelou o interesse do público geral em imagens, o que introduziu o que era conhecido como recuperação de imagens baseada em conteúdo (*content-based image retrieval*). A combinação de arquivos independentes (ou informais), assim como é o caso da *web*, tinha pressa por uma melhor avaliação interativa dos resultados aproximados e, mais adiante, da substituição de algoritmos ineficientes e suas respectivas combinações em classificadores, também ineficientes.

Já no novo século, o último baluarte analógico foi descartado. Em poucos anos, todos os sensores passaram a ser digitais. De acordo com Sebe et al. (2005), arquivos também passarão a ser digitais em breve. Como consequência destas

mudanças nas condições básicas, *datasets* (recursos para armazenamento de dados na memória) sofrerão *overflow* (quantidade de dados maior que a capacidade de armazenamento). A visão computacional irá gerar um novo ramo que será chamado de algo parecido com *archive-based* (baseado em arquivo) ou *semantic vision* (visão semântica), incluindo uma parte para descrição formal de conhecimento em uma ontologia equipada com detectores. Uma visão alternativa seria baseada em experiência (*experience-based*) ou visão cognitiva (*cognitive vision*). Isto é praticamente uma perspectiva de visão dirigida a dados e inclui as leis elementares de formação de imagens.

## 2.1 Definição

De acordo com a definição de Shapiro e Stockman (2001, p.13) “o objetivo da visão computacional é formar decisões úteis sobre objetos físicos reais e cenas baseadas em imagens captadas”. De forma a tomar decisões a respeito de objetos reais, quase sempre se faz necessário construir alguma descrição ou modelo deste objeto a partir da imagem. Por causa disto, muitos especialistas irão dizer que o objetivo da visão computacional é a construção de descrições de cenas a partir de imagens.

Em um ponto de vista bastante semelhante, para Sebe et al. (2005), o objetivo da visão computacional é prover aos computadores uma capacidade de percepção semelhante à humana, de forma que estes possam perceber o ambiente, entender os dados percebidos, tomar ações apropriadas e aprender com estas experiências, de forma a aprimorar desempenhos futuros. Esta área evoluiu a partir da aplicação do reconhecimento de padrões clássico de métodos de processamento de imagens até avançadas técnicas em entendimento de imagens como a visão baseada em modelo e visão baseada em conhecimento, além de outros sistemas com capacidade de aprendizado. A habilidade de entender e de aprender são as duas maiores capacidades associadas com estes sistemas. Nos últimos anos, avanços teóricos e práticos foram realizados no campo da visão computacional e reconhecimento de padrões, com novas técnicas e processos de aprendizagem, representação e adaptação. Seria justo afirmar, entretanto, que a aprendizagem representa a próxima barreira desafiadora à visão computacional.

## 2.2 Componentes de um sistema de visão

A visão computacional é um assunto bastante complexo. Jähne e Haußecker (2000) propõem dividir estes conceitos em componentes ou módulos, de forma a traçar um comparativo entre um sistema técnico e outro biológico, como se pode observar na Tab. 1. Nesta concepção, a funcionalidade comum da visão biológica e de máquina incluiria os seguintes componentes:

- **Fonte de Radiação:** Se nenhuma radiação incide sobre o objeto em estudo, nada se pode observar ou processar. Além disso, uma boa iluminação é necessária para os objetos que não são radiantes por si próprios.
- **Câmera:** Coleta a radiação recebida do objeto, de forma que a radiação original possa ser detectada. Nos casos mais simples, trata-se apenas de lentes ópticas. Em casos mais complexos, poderia ser um sistema mais específico, como um tomógrafo de raios-x, por exemplo.
- **Sensor:** Converte a densidade do fluxo de radiação recebido em um sinal apropriado para um processamento posterior. Em um sistema de imageamento simples, um *array* 2D de sensores é necessário para capturar a distribuição espacial da radiação. Já com um sistema apropriado, apenas um sensor ou uma fileira de sensores é suficiente, em alguns casos.
- **Unidade de Processamento:** Extrai características apropriadas a partir da imagem recebida, de forma a mensurar as propriedades do objeto e classificá-las em classes. Outro componente importante é um sistema de memória para coletar e armazenar conhecimento a respeito da cena em estudo, incluindo mecanismos para apagar coisas sem importância.
- **Atores:** Reagem no resultado da observação. Eles passam a ser parte integral do processo quando o sistema de visão passa a responder ativamente à observação do rastreamento de um objeto de interesse, por exemplo, entre outros casos.

Tabela 1 – Módulos funcionais da visão humana e de máquina.

<b>Tarefa</b>	<b>Visão Humana</b>	<b>Visão de Máquina</b>
Visualização	Passiva, principalmente pela reflexão de luz em superfícies opacas	Passiva e ativa (iluminação controlada), usando eletromagnetismo e radiação acústica
Formação da Imagem	Sistema óptico de refração	Sistemas variados
Controle de irradiação	Pupila controlada por músculos	Aberturas motorizadas, filtros circulares e sintonizáveis
Ajuste do foco	Mudança da distância focal controlada por músculos	Sistemas com foco automático, baseados em medições de distância
Resolução de irradiação	Sensibilidade logarítmica	Sensibilidade linear, quantização entre 8 e 16 <i>bits</i> ; sensibilidade logarítmica
Rastreamento	<i>Eyeball</i> com alta mobilidade	<i>Scanner</i> e câmeras montadas em robôs
Processamento e análise	Processamento paralelo massivo, organizado hierarquicamente	Dominância de processamento serial; processamento paralelo sem uso geral

Fonte: JÄHNE, B.; HAUßECKER, H., 2000.

### 2.3 Aplicações

Segundo Sebe et al. (2005), a demanda por sistemas de visão computacional para aplicações em problemas do mundo real (*“real-world” problems*) têm crescido significativamente. Porém, muitos modelos e metodologias não conseguem fugir de uma espécie de “ambiente de brinquedo” em direção ao mundo real. Portanto, a visão computacional, no estado da arte, representa um significativo

avanço para tratar esses problemas do mundo real, como por exemplo, navegação, reconhecimento de alvos, manufaturas, interpretação de fotos, sensoriamento remoto, entre outros.

É sabido, também, que muitas destas aplicações requerem algoritmos e sistemas e visão que trabalhem sob oclusão parcial, grande quantidade de objetos desorganizados na imagem (*clutter*), baixo contraste e mudanças constantes nas condições do ambiente. Assim, é necessário que as técnicas de visão computacional sejam robustas e flexíveis o suficiente para otimizar a performance do sistema em um determinado cenário.

Como exemplo de aplicação, pode-se citar o uso de visão computacional na Medicina, para processamento de imagens médicas, de forma a auxiliar nos diagnósticos de exames como tomografia, ultrassonografia, microscopia, radiografia, e outros. Outra área de aplicação é a industrial, na qual a visão computacional auxilia em mecanismos de controle de qualidade ou cálculo de posição e orientação de braços robóticos, por exemplo.

Já a área militar é talvez a maior aplicação da visão computacional, embora os trabalhos não sejam disponibilizados ao público em geral. Como exemplos básicos, podem ser citados a detecção de unidades inimigas e mísseis teleguiados. Sistemas mais avançados enviam mísseis para um área ao invés de um alvo específico, sendo que a seleção do alvo é feita no processamento da imagem do local feita pelo próprio míssil.

Uma das novas aplicações são os veículos autônomos, com nível de autonomia variante entre total ou parcial. A autonomia total usa a visão computacional para a navegação, isto é, para obter a localização, para produzir mapas do ambiente e para detectar obstáculos. Já a autonomia parcial auxilia apenas a tarefa de dirigir em situações diversas.

No caso deste trabalho, a aplicação da visão computacional se dá no rastreamento de jogadores a partir de imagens capturadas a partir de vídeos de jogos de esportes coletivos. O rastreamento é outra aplicação da visão computacional que se encontra em um bom nível de progresso atualmente, assunto do próximo capítulo.



## 3 Rastreamento

De acordo com Haritaoglu (1998) e Gabriel et al. (2003), nos últimos anos se iniciou, na área de visão computacional, um movimento que parte do processamento de imagens estáticas em direção ao processamento de seqüências de vídeo. Pesquisas atuais têm investigado o reconhecimento de atividades humanas executadas em cena. Aplicações como base de dados de vídeos, *interfaces* de realidade virtual, sistemas inteligentes de vigilância, videoconferência e aplicações na área esportiva, entre outras, têm em comum o rastreamento e a interpretação destas atividades.

### 3.1 Definição

Segundo Forsyth e Ponce (2002), *Tracking*, ou rastreamento, é o problema de gerar uma inferência a respeito de movimentos de um objeto, a partir de uma dada seqüência de imagens. Boas soluções para este problema possuem várias aplicações, como:

- Captura do movimento: Se é possível rastrear um objeto em movimento de forma precisa, então também é possível armazenar um registro destes movimentos, com precisão. A partir deste registro, pode-se usá-lo para controlar um processo de renderização, de forma a controlar objetos virtuais, gerar cenas com movimentos diferentes dos originais, etc.
- Reconhecimento a partir do movimento: O movimento de cada objeto é extremamente característico. Isto torna possível determinar a identidade do objeto simplesmente a partir de seu movimento ou até mesmo compreender o que este objeto está fazendo.

- **Vigilância:** Saber o que o objeto está fazendo pode ser bastante útil. Existem combinações de locais e movimentos que acontecem freqüentemente, e outras combinações que raramente ocorrem. Estas informações podem ser importantes para sistemas de monitoramento, fornecendo um aviso caso um problema seja detectado.
- **Definição do alvo:** Uma fração significativa da literatura é orientada à decisão de que objeto definir como alvo do rastreamento. Sendo assim, o importante é saber o que se pode inferir a respeito da posição futura de um objeto em uma seqüência de medições.

Desta forma, para Forsyth e Ponce (2002), nos problemas típicos de rastreamento, tem-se um modelo dos movimentos de um objeto e um conjunto de medições a partir de uma seqüência de imagens. Estas medições poderiam ser a posição de alguns pontos da imagem, a posição e momentos de alguma região da imagem ou qualquer outra variável que se deseja medir. Sendo assim, não haveria garantia de relevância, pois poderia tratar-se tanto do objeto de interesse, como de outros objetos ou até mesmo de ruídos presentes nas imagens.

Já Bar-Shalom (2001) define rastreamento como a estimação do estado de um objeto em movimento, baseada em medições remotas. Isto é obtido pela utilização de um ou mais sensores em locais fixos ou plataformas móveis. À primeira impressão, o rastreamento parece ser um caso especial de estimação, mas seu escopo é mais amplo do que isto. Seu processo não só faz uso de todas as ferramentas para estimação, mas também requer um extenso uso de teoria de decisão estatística em sua aplicação prática.

### **3.2 Aplicações**

São exemplos de aplicação de sistemas de rastreamento: segurança e vigilância, qualificação do espaço, controle do tráfego, acessibilidade, entre diversos outros, além de ser um passo fundamental na Visão Computacional, elemento indispensável para a robótica. No caso deste trabalho, a aplicação se dá na área esportiva profissional.

### 3.3 Classes de Métodos de Rastreamento

Segundo Schnack (2005), existem três classes de métodos para o rastreamento de objetos:

- Rastreamento baseado em correspondência: requer a detecção do objeto em toda amostra, sendo executado no estabelecimento da correspondência do objeto nas amostras consecutivas. Consegue-se isto utilizando uma modelagem a partir do estado do objeto, no qual as variáveis de estado podem ser compostas de um modelo linear de movimento (velocidade e aceleração constantes), unidas com informações da região (cor, textura e forma). Baseada no modelo, a correspondência do objeto é feita através da predição da nova posição do objeto usando observações passadas e verificando a existência do objeto na posição prevista.
- Rastreamento baseado no contorno: requer a detecção do objeto apenas uma vez. O rastreamento é feito achando o contorno do objeto dado o contorno na amostra anterior. Essa identificação do contorno é efetuada usando a gradiente da imagem.
- Rastreamento baseado em transformação: requer a detecção do objeto apenas uma vez. O rastreamento é atingido estimando a movimentação do objeto nos quadros (*frames*) consecutivos. Essa estimativa do movimento do objeto pode ser feita com cálculos baseado nas informações do objeto (cor, textura, forma, etc.), com filtros de Kalman e/ou combinações dessas técnicas para um melhor desempenho. Uma técnica bastante difundida para esse tipo de rastreamento é a do vetor *mean shift*, empregado no sistema Kinesis, que utiliza o histograma de cores do objeto para identificar sua translação nas amostras consecutivas, conforme será explicado com detalhes no capítulo 4.

Ao escolher o método de rastreamento para o sistema Kinesis, Schnack (2005), levou em conta duas características fundamentais na escolha da técnica de rastreamento a ser usada. A primeira é que o rastreamento será feito em vídeos de jogos de esportes coletivos com atletas de alto nível, ou seja, a movimentação desses jogadores será bem acentuada e rápida. Assim sendo, a forma do corpo dos

atletas em algumas partes do vídeo será deformada pela velocidade do movimento. A outra característica desejada é a de que o sistema opere em tempo real, de forma a extrair as métricas durante a partida e proporcionar dados para o avaliador no exato momento em que as ações ocorrem.

Dessa forma, a questão do tempo real exclui a possibilidade de se utilizar o rastreamento baseado em correspondência, visto que a necessidade de detecção do objeto em todas as amostras inviabiliza tal possibilidade. A chance de se utilizar o rastreamento baseado em contorno também foi reduzida, de uma maneira mais sutil, pela questão de ser computacionalmente mais complexa que a técnica baseada em transformação.

Entretanto, o motivo principal para a não utilização do rastreamento baseado em contorno foi que os jogadores podem se deformar em certas amostras, devido ao seu rápido movimento. Assim sendo, o tipo de rastreamento mais apropriado para o objetivo do sistema Kinesis, é o baseado em transformação, utilizando a técnica do *mean shift*, pois ela permite tratar da questão da deformação e seu custo computacional é baixo a ponto de poder ser executado em tempo real.

Com base nisto, este trabalho busca aplicar outra técnica de rastreamento baseado em transformação, o filtro de Kalman, como apoio ou em conjunto ao *mean shift*, buscando uma melhora no rastreamento do Kinesis frente às situações de oclusão e deformação dos alvos. Esta técnica é apresentada no capítulo 5 deste trabalho.

### **3.3 Revisão Bibliográfica sobre Sistemas de Rastreamento**

Seguindo outra linha de classificação, Misuta (2004), apresentou uma revisão bibliográfica de sistemas de rastreamento de jogadores de futebol, classificada de acordo com sua base de aquisição de dados. Os artigos publicados em revistas e anais dos congressos nacionais e internacionais que foram consultados abrangem o período de 1976 a 2003. As bibliotecas eletrônicas consultadas foram Pubmed e Probe com as palavras chave: *tracking players, soccer, game analysis, motion analysis, global positional systems (GPS), soccer actions, quantification of games, movement speeds and distances covered*. Os trabalhos encontrados baseiam-se nos seguintes métodos: método baseado em

estimativa visual, métodos baseados em sensores-transmissores e métodos baseados em imagens.

De acordo com Misuta (2004), o método baseado em estimativa visual foi um passo importante como uma forma de obtenção de trajetórias de jogadores de futebol. Porém, trata-se de um método impreciso e demorado. Os trabalhos encontrados realizaram análise de padrões de movimentos dos jogadores ou outras características, como velocidade e aceleração. Em dois deles, o comprimento médio de uma passada para as diferentes situações (caminhar, trotar, correr, entre outras) era conhecido. O terceiro trabalho utilizou um aparelho de vídeo-cassete, uma folha quadriculada adaptada ao monitor de TV e uma folha quadriculada maior na qual eram transcritos os pontos de correspondência relacionados a um determinado jogador. Estes dados podem ser conferidos na Fig. 1.

<b>MÉTODOS BASEADOS EM ESTIMATIVA VISUAL</b>				
<b>Título</b>	<b>Autor</b>	<b>Objetivo</b>	<b>Metodologia</b>	<b>Resultados</b>
A Motion Analysis of Work-rate in Different Positional Roles in Professional Football Match-play	Reilly e Thomas (1976)	Análise de padrões de movimentos dos jogadores	Jogos da 1ª divisão do campeonato inglês. Estimativa da distância percorrida por um jogador durante todo o jogo conhecendo o comprimento médio de uma passada nas seguintes situações (andar, andar de costas, trotar, correr e "sprint") do atleta a ser analisado.	Obtenção da distância percorrida de um jogador para o jogo todo.
Match Analysis of Australian Professional Soccer Players	Withers et al. (1982)	Análise de padrões de movimentos dos jogadores	Jogos da 1ª divisão do campeonato australiano. Estimativa da distância percorrida por um jogador durante todo o jogo conhecendo o comprimento médio de uma passada nas seguintes situações (caminhar, trotar, correr, "sprint", movimento lateral, andar para trás e trotar para trás) do atleta a ser analisado.	Obtenção da distância percorrida de um jogador para o jogo todo.
Quantification of Games - Preliminary Kinematic Investigations in Soccer	Erdmann (1991)	Rastreamento da trajetória e análise de velocidade e aceleração	Jogo de futebol da terceira divisão; Utilizou um aparelho de vídeo-cassete, uma folha quadriculada adaptada ao monitor de TV e uma folha quadriculada maior na qual eram transcritos os pontos de correspondência relacionados a um determinado jogador(es).	Obtenção da distância percorrida e análise da distribuição de velocidade e aceleração para um jogador durante 5 minutos de jogo.

Figura 1: Síntese dos Trabalhos Relativos aos Métodos Baseados em Estimativa Visual.

Fonte: MISUTA, 2004, p. 12.

Já os métodos baseados em sensores-transmissores possibilitam a obtenção da posição em função do tempo de maneira automática, porém levando em conta as dimensões do aparelho, podem acarretar risco ao atleta. Além disto, a

sua aplicação nem sempre é permitida devido às restrições das regras do esporte. Ambos os trabalhos encontrados utilizaram a tecnologia GPS em atletas submetidos a testes para avaliação de características como velocidade, aceleração e distância percorrida, identificando, no primeiro trabalho, o tipo de movimento através do erro apresentado, enquanto no segundo trabalho foram comparados os dados para jogadores de futebol que atuam em diferentes posições em campo, conforme mostra a Fig. 2.

<b>MÉTODOS BASEADOS EM SENSORES-TRANSMISSORES</b>				
<b>Título</b>	<b>Autor</b>	<b>Objetivo</b>	<b>Metodologia</b>	<b>Resultados</b>
The use of global positioning systems (GPS and DGPS) for the tracking of human motion	Hennig e Sterzing (1999)	Desenvolvimento de um método para rastreamento de trajetórias	Rastreamento de 6 voluntários em uma pista de 400m utilizando o sistema GPS. Cada voluntário foi orientado a percorrer uma volta completa nos seguintes tipos de movimentos: andar, correr e "sprint"	Os erros percentuais médios para a distância percorrida apresentaram os seguintes resultados: a) erros menores que 2%: correr e "sprint"; b) erros de 10%: andar.
Game analysis by GPS satellite tracking of soccer players	Hennig e Briehle (2000)	Rastreamento automático de trajetórias	Foram analisados 70 jogadores de futebol de diversas categorias alemãs em 14 jogos; Os estudos foram feitos nos treinos destas equipes durante 90 minutos; Utilizou o GPS e um aparelho receptor GARMIN 12XL GPS.	Obteve a distância percorrida e a velocidade para 90 minutos num jogo durante o treino para um defensor, lateral, meio-campista e atacante;

Figura 2: Síntese dos Trabalhos Relativos aos Métodos Baseados em Sensores-Transmissores.

Fonte: MISUTA, 2004, p. 12.

Alguns dos trabalhos baseados em imagens utilizaram uma curta seqüência de imagens para desenvolver métodos visando solucionar os problemas de oclusão mútua entre jogadores, identificação de jogadores de times diferentes e técnicas de rastreamento automático utilizando principalmente o filtro de Kalman. Outra técnica aplicada em alguns trabalhos foi a de triangulação, a partir de valores de ângulos obtidos por goniômetros (qualquer instrumento empregado para medir ângulos). Estas informações podem ser conferidas na Fig. 3.

O presente trabalho, seguindo a classificação proposta por Misuta (2004), utiliza métodos baseados em imagens. O filtro de Kalman e o algoritmo de *mean shift* são alguns exemplos destas técnicas empregadas no sistema Kinesis.

<b>MÉTODOS BASEADOS EM IMAGENS</b>				
<b>Título</b>	<b>Autor</b>	<b>Objetivo</b>	<b>Metodologia</b>	<b>Resultados</b>
Measuring Movement speeds and Distances Covered During Soccer Match-Play	Ohashi et al. (1987)	Rastreamento da trajetória jogadores	Jogos da Liga Japonesa de Futebol. Obtenção da posição em função do tempo de um jogador de cada vez utilizando a técnica de triangulação para estimar a posição dos jogadores a partir de valores de ângulos obtidos por goniômetros.	Obtenção da distância percorrida e análise da distribuição de velocidades de um jogador para o jogo todo. Foram analisados 4 jogadores ao todo.
El rendimiento Del jugador de fútbol	D'Ottavio e Tranquilli (1993)	Rastreamento da trajetória de jogadores de futebol e análise de velocidade e aceleração	Obtenção da posição em função do tempo de um jogador utilizando a técnica de triangulação para estimar a posição dos jogadores a partir de valores de ângulos obtidos por goniômetros.	Obtenção da distância percorrida e análise da distribuição de velocidades de um jogador para o jogo todo.
Visual tracking using closed-worlds	Intille e Bobick (1994)	Desenvolvimento de um método para rastreamento automático de trajetórias e identificação de jogadas	Jogo de futebol americano; método denominado "Closed-Worlds". Sequência de imagens com 600 quadros.	O método obteve sucesso nos casos em que não houve oclusão e correspondeu a 400 quadros.
Where are the ball and players? Soccer game analysis with color-based tracking and image mosaick	Choi et al. (1997)	Desenvolvimento de um método para rastreamento automático de trajetórias	Jogo de futebol. Sequência de imagens de TV a cabo com 5 segundos; Método denominado "analysis with color-based tracking and image mosaick". O rastreamento utiliza a predição baseada no filtro de Kalman.	Identificação dos jogadores e rastreamento nos 5 segundos nas situações sem oclusão entre jogadores.
Tracking multiple sports players through occlusion, congestion and scale	Needham e Boyle (2001)	Desenvolvimento de um método para rastreamento automático de trajetórias	Jogo de futebol de salão. Método denominado "multiple object Condensation"; Sequência de imagens com 835 quadros; Utiliza a predição baseada no filtro de Kalman para fazer o rastreamento.	Segundo o autor, 56% da trajetória obtida na medição automática correspondem à trajetória realizada pelo atleta.

Figura 3: Síntese dos Trabalhos Relativos aos Métodos Baseados em Imagens.

Fonte: MISUTA, 2004, p. 13.

## 4 Kinesis e *Mean Shift*

Iniciado sob orientação do Prof. Dr. Dante Barone, no Laboratório de Robótica Inteligente da Universidade Federal do Rio Grande do Sul (UFRGS), o Kinesis tem o objetivo final de efetuar o rastreamento de jogadores de esportes coletivos em tempo real. Ainda em fase de desenvolvimento e estudos, o protótipo do Kinesis disponibilizou, em sua primeira versão, um ambiente capaz de extrair dados a partir de vídeos de jogos, de forma a auxiliar profissionais da área esportiva.

A primeira versão do Kinesis, documentada por Schnack (2005), é apresentada resumidamente neste capítulo, juntamente com suas características básicas, em especial o algoritmo de *mean shift* (MS), base do rastreamento do sistema. Também foram acrescentadas, neste capítulo, informações adicionais às de Schnack (2005), relevantes para o presente trabalho, que se trata de uma continuação do projeto inicial do Kinesis. O foco do texto é tratar os conceitos de Visão Computacional e Processamento de Imagens empregados no sistema como objetos de estudo e discussão.

### 4.1 Mean Shift

A estimação do gradiente de uma função de densidade, conhecida por *mean shift*, foi desenvolvida por Fukunaga e Hostetler (1975). Mais recentemente, as propriedades de compressão de dados e redução das dimensões do MS têm sido exploradas em tarefas de baixo nível na área de visão computacional. Trabalhos como análise de relações espaciais, por Comaniciu e Meer (2002), rastreamento de corpos humanos, desenvolvido por Porikli e Tuzel (2003) e rastreamento de objetos em movimento a partir de vídeos, por Nedović (2004) são exemplos de aplicação do algoritmo de *mean shift* neste tipo de tarefa.



De acordo com Comaniciu e Meer (1999), o MS pode ser definido como um estimador não-paramétrico de densidade de gradiente e é comumente empregado como filtro de preservação de descontinuidade e segmentação de imagens, ao longo do domínio espacial de cores e níveis de cinza. O método de filtragem proposto no desenvolvimento do Kinesis associa *pixels* das imagens com regiões semelhantes na distribuição de densidade do domínio ao qual o filtro é empregado. Essa similaridade é expressa no sistema por uma métrica baseada no coeficiente de Bhattacharyya, explicado mais adiante neste capítulo.

Já Ukrainitz e Sarel (2004) propõem uma descrição intuitiva a partir de um exemplo simples. Supõe-se que, a partir de uma distribuição de bolas de bilhar idênticas, escolha-se uma região de interesse qualquer, conforme mostrado na Fig. 4. O algoritmo, através da utilização de um histograma de cores, detecta a região com maior densidade de pontos de uma determinada cor e calcula uma média ponderada para se achar o centro de massa desta região. Logo após é traçado um vetor entre o centro da região de interesse e o centro de massa obtido, conhecido como vetor de *mean shift*.

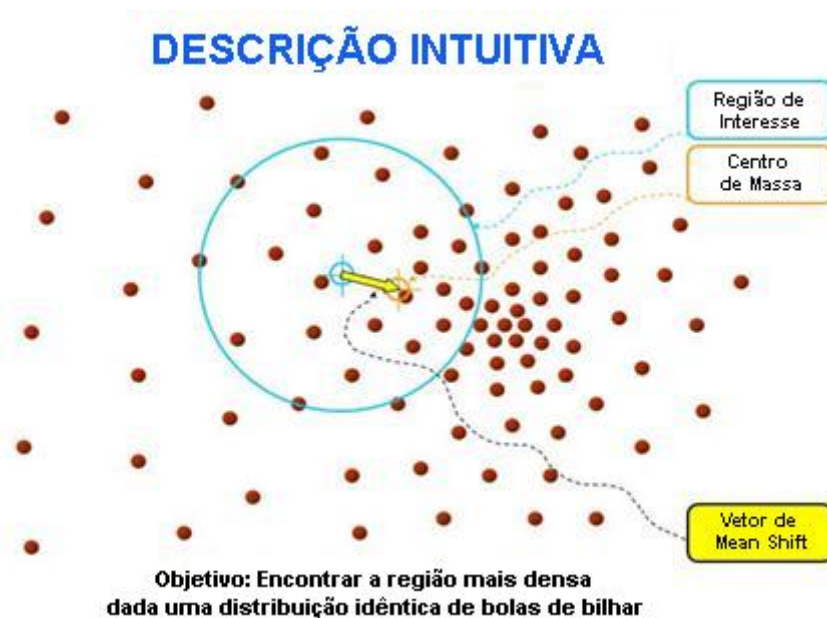


Figura 4: Região de interesse, centro de massa e vetor *mean shift*.

Fonte: UKRAINITZ, Y; SAREL, B., 2004.

A seguir, a região de interesse inicial é transladada até que seu centro coincida com o centro de massa, atingindo uma convergência (Fig. 5). Este processo

se repete, até que não se encontre regiões com maior densidade do que a região já obtida.

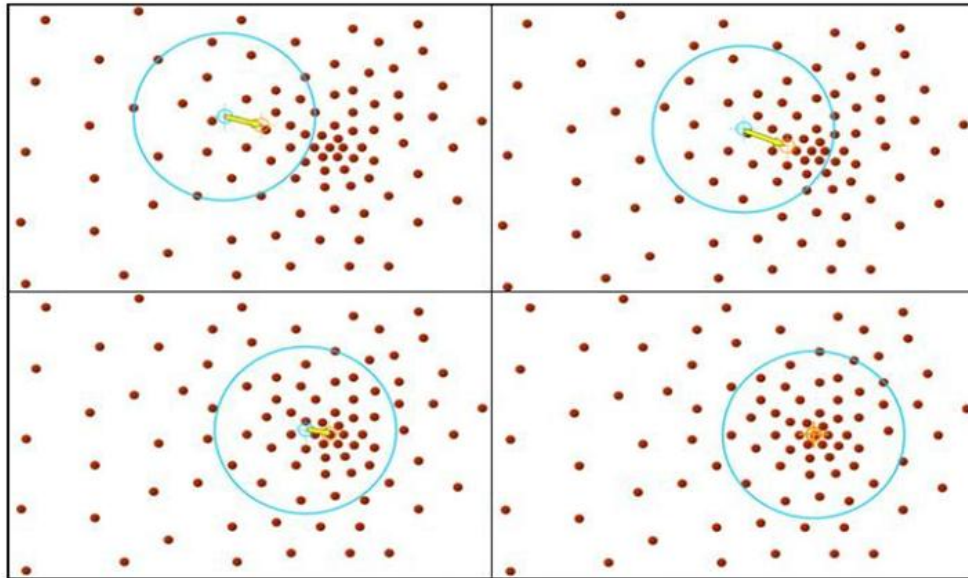


Figura 5: Convergência no procedimento de *mean shift*.

Fonte: UKRAINITZ, Y; SAREL, B., 2004.

O *mean shift* trata-se de um processo vantajoso, pois não é necessária uma avaliação da imagem por inteiro, de forma que se ganha em rapidez.

## 4.2 Função Densidade de Probabilidade

Uma Função Densidade de Probabilidade (*Probability Density Function - PDF*), de acordo com Archambeau e Verleysen (2003), oferece uma forma flexível de investigar propriedades de um conjunto de dados, de forma a prover uma base sólida para mineração de dados (*data mining*). Por outro lado, uma PDF é crucial em tarefas de aprendizado de máquina não-supervisionado e inferência e classificação Bayesiana.

Para a estimação de densidade, três alternativas podem ser consideradas. São elas: a estimativa paramétrica, semi-paramétrica e não-paramétrica de densidade, sendo a última o tipo de estimação realizada pelo *mean shift*.

Segundo Ukrainitz e Sarel (2004), em uma estimativa não paramétrica de densidade, as probabilidades da função de densidade são extraídas diretamente do

conjunto de dados reais (Fig. 6). Assim, as regiões mais densas correspondem a máximos da função de probabilidade e as menos densas representam os mínimos. Ou seja, quanto mais densa a região, maior a probabilidade na função probabilística de densidade, conforme mostrado na Fig. 7.

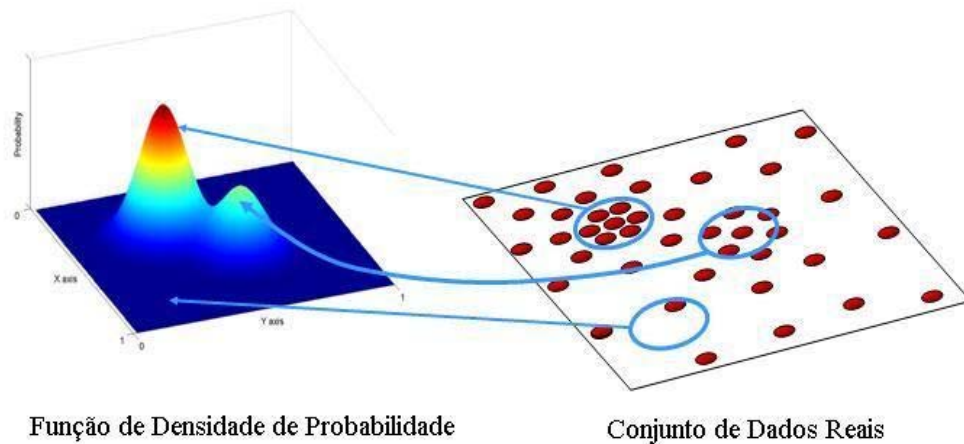


Figura 6: Extração de dados de um conjunto real.

Fonte: UKRAINITZ, Y; SAREL, B., 2004.

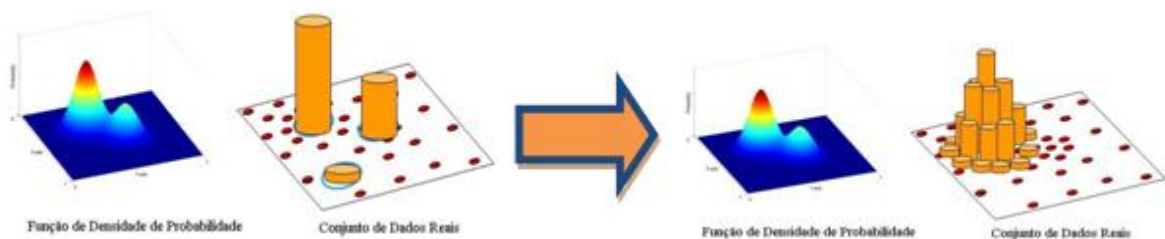


Figura 7: Estimação da PDF.

Fonte: UKRAINITZ, Y; SAREL, B., 2004.

#### 4.2.1 Estimativa de Densidade de *Kernel*

De acordo com Comaniciu e Meer (2000), o objetivo da estimativa de densidade é aproximar a função de probabilidade de densidade de uma variável randômica  $x$ . Assumindo  $n$  observações independentes  $x_1, \dots, x_n$  da variável randômica  $n$ , o estimador de densidade de *kernel* para o valor da densidade  $f(x)$  no ponto  $x$  é definido por:

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (1)$$

sendo  $K(\cdot)$  chamado de função de *Kernel* e  $h$  a largura de banda.

A função de *kernel*  $K(\cdot)$  pode ser construída a partir da rotação do perfil do *kernel*  $k(\cdot)$  em  $R^d$ . Assim sendo,  $K(\cdot)$  é radialmente simétrico:

$$K(x) = c_{k,d} k(\|x\|^2) \quad (2)$$

onde  $c_{k,d}$  é necessário para garantir a normalização do perfil.

Utilizando a notação de perfil, podemos reescrever a equação (1) da seguinte forma:

$$\hat{f}_{h,K}(x) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^n k\left(\left\|\frac{x - x_i}{h}\right\|^2\right) \quad (3)$$

Segundo Ferrugem et al. (2004), a qualidade da estimativa de densidade do *kernel* é medida pela média do erro quadrado entre a densidade e a estimativa da mesma. O *kernel* que minimiza tal medida é o Epanechnikov, cujo perfil é

$$k_E(x) = \begin{cases} 1 - x & \text{se } 0 \leq x \leq 1 \\ 0 & \text{se } x > 1 \end{cases} \quad (4)$$

e gera o *kernel* radialmente simétrico

$$K_E(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d + 2) (1 - \|x\|^2) & \\ 0 & \end{cases} \quad (5)$$

onde  $c_d$  é o volume da esfera d-dimensional.

O perfil de Epanechnikov, conforme mostrado na Fig. 8, tem influência direta na construção dos histogramas das imagens com as quais se está trabalhando. Ele associa pesos menores aos *pixels* mais afastados do centro, e maiores aos mais próximos. Desta forma, quanto mais afastado das coordenadas do centro, menos confiáveis são os *pixels*, podendo pertencer ao fundo da imagem ou a outros objetos que estejam causando oclusão sobre o alvo original do rastreamento.

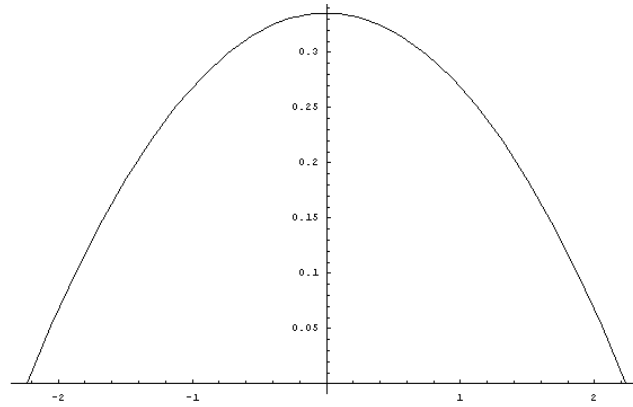


Figura 8: Perfil de Epanechnikov.

Fonte: WIKIPEDIA, 2007.

Como dito anteriormente, o *mean shift* é uma ferramenta para localizar modas em conjuntos de dados, a partir de uma densidade  $f(x)$ . As modas localizam-se onde o gradiente  $\nabla f(x) = 0$  e o MS localiza tais zeros sem estimar a densidade.

#### 4.2.2 Estimativa do Gradiente da Densidade

De acordo com Ferrugem et al. (2004), a partir da linearidade de (3), pode-se estimar o gradiente da densidade como:

$$\hat{\nabla}f_{h,K}(x) \equiv \nabla \hat{f}_{h,K}(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^n (x - x_i) k' \left( \left\| \frac{x - x_i}{h} \right\|^2 \right) \quad (6)$$

A função  $g(x) = -k'(x)$  é definida assumindo que a derivada do perfil  $k$  do *kernel* existe para todos os pontos  $x \in [0, \infty)$ , exceto por um conjunto finito de pontos, segundo Comaniciu e Meer (2000). Utilizando  $g(x)$  como perfil, pode-se definir o *kernel*  $G(x)$  como:

$$G(x) = c_{g,d} g(\|x\|^2) \quad (7)$$

onde  $c_{g,d}$  é a constante de normalização correspondente. O *kernel* uniforme é a derivada do *kernel* de Epanechnikov e o *kernel* normal é a sua própria derivada.

Introduzindo  $g(x)$  em (6), temos:

$$\widehat{\nabla} f_{h,K}(x) = \frac{2c_{k,d}}{nh^{d+2}} \left[ \sum_{i=1}^n g \left( \left\| \frac{x - x_i}{h} \right\|^2 \right) \right] \left[ \frac{\sum_{i=1}^n x_i g \left( \left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^n g \left( \left\| \frac{x - x_i}{h} \right\|^2 \right)} - x \right] \quad (8)$$

A partir de (6), pode-se deduzir que o primeiro termo do produto de (8) é proporcional à estimativa de densidade em  $x$  computada com o *kernel*  $G$  (9).

$$\hat{f}_{h,G}(x) = \frac{c_{g,d}}{nh^d} \sum_{i=1}^n g \left( \left\| \frac{x - x_i}{h} \right\|^2 \right) \quad (9)$$

e o segundo termo é o *mean shift* (10)

$$m_{h,G}(x) = \left( \frac{\sum_{i=1}^n x_i g \left( \left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^n g \left( \left\| \frac{x - x_i}{h} \right\|^2 \right)} \right) - x \quad (10)$$

isto é, a diferença entre a média ponderada, utilizando o *kernel*  $G$  para os pesos, e  $x$  o centro do *kernel*. Desta forma, a partir de (9) e (10), pode-se reescrever (8) como:

$$\widehat{\nabla} f_{h,K}(x) = \hat{f}_{h,G}(x) \frac{2c_{k,d}}{h^2 c_{g,d}} m_{h,G}(x) \quad (11)$$

obtendo:

$$m_{h,G}(x) = \frac{1}{2} h^2 c \frac{\widehat{\nabla} f_{h,K}(x)}{\hat{f}_{h,G}(x)} \quad (12)$$

A equação (12) mostra que, em uma localização  $x$ , o vetor de *mean shift* computado com *kernel*  $G$  é proporcional à estimativa do gradiente de densidade normalizada, obtida com *kernel*  $K$ . A normalização é realizada a partir da estimativa da densidade em  $x$ , computada com o *kernel*  $G$ . Assim, o vetor de *mean shift* sempre aponta para a direção do acréscimo máximo na densidade.

Disto infere-se que o MS possui velocidade de convergência automática, pois o tamanho do seu vetor depende do gradiente. Assim sendo, perto dos valores máximos, os passos são curtos e refinados. Uma vantagem desta técnica é a de que se pode lidar com qualquer espaço arbitrário e escolher um único parâmetro.

### 4.3 Métrica Baseada no Coeficiente de Bhattacharyya

O *mean shift*, ao longo de suas iterações, busca o candidato mais similar ao modelo do objeto a ser rastreado. Essa similaridade é expressa no sistema Kinesis por uma métrica baseada no coeficiente de Bhattacharyya.

Comaniciu e Meer (2000) afirmam que a tarefa de localizar um alvo é feita da seguinte forma: tem-se uma característica  $z$  representando a cor ou textura do modelo do alvo, cuja densidade é representada por  $q_z$ , enquanto o candidato do alvo centrado em uma posição  $y$  tenha a característica distribuída de acordo com  $p_z(y)$ . O problema é, então, encontrar a localização discreta de  $y$ , cuja densidade associada  $p_z(y)$ , é a mais similar à densidade do alvo  $q_z$ .

Segundo Ferrugem et al. (2004), para definir uma média de similaridade, leva-se em conta a relação direta da probabilidade de erro na classificação em hipóteses estatísticas com a semelhança entre as duas distribuições. Quanto maior a probabilidade de erro, mais similares as distribuições serão. Desta forma, o problema da estimativa da localização do alvo é formulado como a estimativa que maximiza o erro bayesiano associado com as distribuições do modelo e do candidato. Uma entidade bem relacionada com o erro bayesiano é o coeficiente de Bhattacharyya, cuja forma geral é dada por:

$$\hat{\rho}(y) \equiv \rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(y) \hat{q}_u} \quad (13)$$

A derivação do coeficiente de Bhattacharyya de certos dados envolve a estimativa das densidades  $p$  e  $q$ , para a qual é utilizada a formulação de histogramas. Apesar de não ser a melhor estimativa de densidade não-paramétrica, o histograma satisfaz o baixo custo computacional requerido para o objetivo final do Kinesis, de realizar processamento em tempo real, de acordo com Schnack (2005).

A densidade do modelo do alvo  $\hat{q} = \{\hat{q}_u\}_{u=1\dots m}$  (com  $\sum_{u=1}^m \hat{q}_u = 1$ ) é estimada através do histograma, com  $m$  bins, enquanto  $\hat{p}$  é estimado em uma determinada localização  $y$  a partir do histograma, com  $m$  bins,  $\hat{p}(y) = \{\hat{p}_u(y)\}_{u=1\dots m}$  (com  $\sum_{u=1}^m \hat{p}_u = 1$ ) do candidato do alvo. Assim sendo o coeficiente de Bhattacharyya é dado por (13).

De acordo com Comaniciu, Ramesh e Meer (2003), a interpretação geométrica de (13) corresponde ao cosseno do ângulo formado pelos vetores unidades m-dimensional  $(\sqrt{\hat{p}_1}, \dots, \sqrt{\hat{p}_m})^T$  e  $(\sqrt{\hat{q}_1}, \dots, \sqrt{\hat{q}_m})^T$ . Desta forma, a distância entre as duas distribuições pode ser definida como:

$$d(y) = \sqrt{1 - \rho[\hat{p}(y), \hat{q}]}$$
 (14)

A Fig. 9 ilustra o gráfico da variação do coeficiente de Bhattacharyya por *frames*, obtido em um dos testes iniciais do sistema Kinesis, realizado por Schnack (2005). A situação em questão era de oclusão total do atleta escolhido como alvo do rastreamento. Pode-se notar que o coeficiente mantém um valor médio em grande parte do tempo, mas apresenta duas quedas drásticas, que correspondem a dois momentos de oclusão no vídeo.

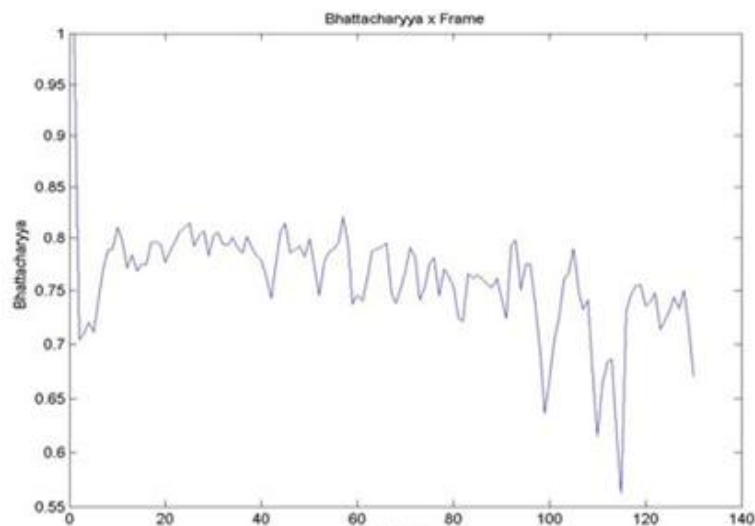


Figura 9: Gráfico do coeficiente de Bhattacharyya para teste de oclusão total.

Fonte: SCHNACK, M., 2005.

## 4.4 Representação do Alvo

### 4.4.1 Modelo do Alvo

Segundo Comaniciu e Meer (2000), para ajustar o modelo do alvo deve-se atribuir  $\{x_i^*\}_{i=1\dots n}$  à localização dos *pixels* do modelo do alvo, centrado em 0. A seguir, é definida uma função  $b: R^2 \rightarrow \{1 \dots m\}$ , que mapeia o *pixel* na localização  $x_i$



para o índice  $b(x_i)$  do *bin* do histograma correspondente à cor desse *pixel*. A probabilidade da cor  $u$  no modelo do alvo é derivada da utilização de um perfil de *kernel* decrescente monotônico  $k$ , que associa pesos menores a *pixels* mais distantes do centro do alvo. O fato de utilizar pesos aumenta a robustez da estimativa, pois os *pixels* mais periféricos são menos confiáveis, visto que, geralmente, pertencem ao fundo ou são afetados por oclusão. O raio do *kernel* é considerado 1, assumindo que as coordenadas  $x$  e  $y$  são normalizadas por  $h_x$  e  $h_y$ , respectivamente. Então, é possível escrever:

$$\hat{q}_u = C \sum_{i=1}^n k(\|x_i^*\|^2) \delta[b(x_i^*) - u] \quad (15)$$

onde  $\delta$  é a função delta de Kronecher. A constante  $C$  de normalização é necessária por causa da condição imposta de  $\sum_{u=1}^m \hat{q}_u = 1$ , de onde

$$C = \frac{1}{\sum_{i=1}^n k(\|x_i^*\|^2)} \quad (16)$$

#### 4.4.2 Candidato do Alvo

De acordo com Comaniciu e Meer (2000), deve-se atribuir  $\{x_i\}_{i=1\dots n_h}$  à localização dos *pixels* do candidato ao alvo, centrado em  $y$  no *frame* atual. Utilizando-se o mesmo perfil de *kernel*  $k$ , porém com raio  $h$ , a probabilidade da cor  $u$  no candidato do alvo é representada por:

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \quad (17)$$

onde  $C_h$  é a constante de normalização. O raio do perfil do *kernel* determina o número de *pixels* do candidato do alvo. Impondo a condição de que  $\sum_{u=1}^m \hat{p}_u = 1$ , obtém-se

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)} \quad (18)$$

É importante observar que  $C_h$  não depende de  $y$ , sendo que as localizações  $x_i$  dos *pixels* estão organizadas em uma forma regular, com  $y$  sendo um dos nodos da forma. Assim,  $C_h$  pode ser calculado previamente para um dado *kernel* e valor  $h$ .

## 4.5 Representação de Cores

De acordo com Jähne (2002), a retina do olho humano, onde imagens são projetadas, contém duas classes de receptores: os cones e os bastonetes. Fotopigmentos nos outros segmentos externos dos receptores absorvem radiação. A energia absorvida é então convertida em sinais eletroquímicos nervosos, os quais são transmitidos via neurônios e nervo óptico para o cérebro. Três diferentes tipos de fotopigmentos presentes nos cones os tornam sensíveis às diferentes larguras do espectro e, desta forma, possibilitam a visão das cores. A visão através dos cones é ativa apenas com média ou alta iluminação, enquanto a visão através dos bastonetes trabalha com níveis baixos de iluminação. Existem três tipos de cones no olho humano, sendo cada um capaz de distinguir um comprimento de onda diferente, correspondentes às cores verde, vermelho ou azul. A quantidade de cones varia para cada cor.

Desta forma, o sistema de cores utilizado basicamente pela visão humana é o modelo RGB (*Red*, *Green* e *Blue*). Além deste sistema de cor existem diversos outros, sendo os mais conhecidos os modelos CMYK (*Cyan*, *Magenta*, *Yellow* e *Black*) e HSB (*Hue*, *Saturation* e *Brightness*). O sistema Kinesis utiliza o sistema HSB, convertido a partir de imagens capturadas de vídeos, com sistema RGB, e amplamente utilizado em problemas de Visão Computacional. Portanto, será dado foco a uma explicação um pouco mais detalhada apenas destes dois sistemas de cor, e da conversão do sistema RGB para o sistema HSB.

### 4.5.1 Sistema RGB

Segundo Jähne (2002), o modelo de representação de cores RGB consiste no uso de três cores primárias aditivas: vermelho, verde e azul. A combinação destas cores de forma aditiva gera a cor resultante.

O espectro do modelo RGB pode ser representado como um cubo tridimensional com as componentes nos cantos de cada eixo, conforme mostrado por Bigun (2006) na Fig. 10. A cor preta se encontra na origem e a cor branca, no canto oposto do cubo. A escala de cinza segue a linha traçada entre o preto e o branco. Em um sistema gráfico de 24 bits, com 8 bits por componente, que é o mais utilizado entre os sistemas RGB, uma cor pura é representada, por exemplo, como (0,255,0) para o verde, correspondendo a (0,1,0) no cubo.

De acordo com Schnack (2005), este modelo simplifica o design de sistemas gráficos, porém não é o ideal para todas as aplicações. Isto se deve ao fato de os componentes vermelho, verde e azul serem amplamente correlacionados, o que dificulta a execução de certos algoritmos de processamento. Diversas técnicas de processamento, como a equalização do histograma trabalham apenas na informação da intensidade de imagem, portanto são mais facilmente implementados utilizando um modelo que já contenha essa informação como, por exemplo, o HSB.

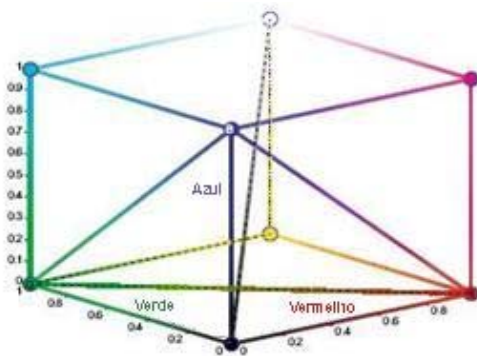


Figura 10: Sistema de Cores RGB.

Fonte: BIGUN, J., 2006.

#### 4.5.2 Sistema HSB

De acordo com Bigun (2006), o modelo HSB trabalha com três valores: Matiz, Saturação e Brilho (*Hue, Saturation e Brightness*). Trata-se de uma descrição de cores baseada no contraste e, portanto, mais intuitiva. Normalmente é representada por um sistema 3D de coordenadas polares, como pode ser visto na Fig. 11.

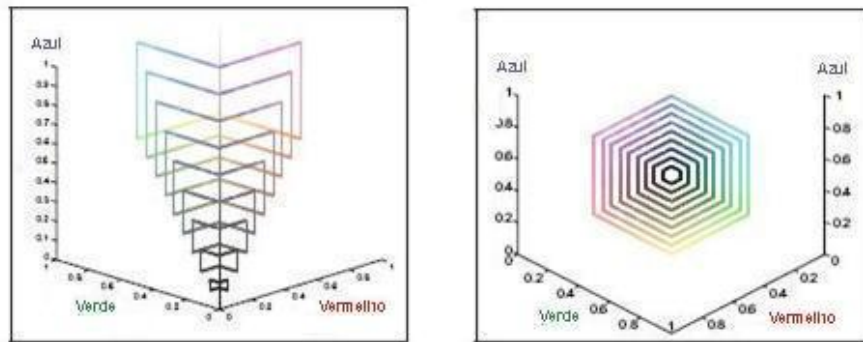


Figura 11: Coordenadas do sistema HSB, com saturação 0,5.

Fonte: BIGUN, J., 2006.

Segundo Forsyth e Ponce (2002), a matiz varia entre 0 (vermelho), passando pelo laranja, amarelo, verde, azul, púrpura, e 359 (novamente vermelho), a cada 60 graus. O brilho varia de 0 a 1 (preto e branco), sendo que as cores puras se encontram no plano 0,5. Já a saturação varia de 0 a 1. Quando o brilho está em 0,5 e a saturação em 1 tem-se as cores puras. Quanto menor a saturação menos pura é a cor.

A escolha da melhor representação de cores para o sistema Kinesis foi feita levando-se em conta sua aplicação prática, ou seja, a avaliação de jogos de esportes coletivos, principalmente o futebol. O ambiente utilizado pelos atletas será constantemente afetado pela luminosidade. Assim sendo, os jogadores a se rastrear terão a intensidade de suas cores afetada pela modificação da iluminação, ou seja, haverá momentos em que o jogador estará com todo o corpo exposto à fonte de iluminação, momentos em que parte do seu corpo estará à sombra de outro jogador ou de outros objetos, e momentos em que todo o seu corpo estará na sombra. Portanto, essas alterações farão com que o mesmo jogador possa ter histogramas bastante diferentes para quadros (*frames*) distintos.

Desta forma, é interessante poder retirar o efeito da mudança na intensidade na representação de cores. A representação RGB sofre alterações nos seus três componentes caso a luminosidade seja modificada, enquanto a representação HSB sofre alteração significativa no seu componente de intensidade, mas seu componente de matiz não sofre modificação perceptível. Com base nessa possibilidade de atenuação do problema de luminosidade inconstante, a representação HSB foi utilizada no sistema Kinesis. Para a construção do histograma do modelo do alvo e de seus candidatos foram levadas em conta as

duas componentes menos afetadas pela variação da luminosidade: matiz e saturação.

### 4.5.3 Conversão RGB para HSB

Conforme descrito por Begun (2006), a conversão do modelo RGB para HSB é realizada a partir das seguintes equações (assumindo  $MAX$  como o valor máximo da tripla  $(R, G, B)$  e  $MIN$  como o menor valor entre eles):

$$H = \begin{cases} 60 \times \frac{G - B}{MAX - MIN} + 0, & \text{se } MAX = R \text{ e } G \geq B \\ 60 \times \frac{G - B}{MAX - MIN} + 360, & \text{se } MAX = R \text{ e } G < B \\ 60 \times \frac{B - R}{MAX - MIN} + 120, & \text{se } MAX = G \\ 60 \times \frac{R - G}{MAX - MIN} + 240, & \text{se } MAX = B \end{cases} \quad (19)$$

$$S = \frac{MAX - MIN}{MAX} \quad (20)$$

$$B = MAX \quad (21)$$

### 4.6 Representação do Fundo

Schnack (2005), no projeto da primeira versão do Kinesis, optou por não utilizar técnicas de subtração de fundo para o sistema Kinesis, por duas razões. Uma delas é no caso de alguma característica do alvo também estar presente no fundo, o que diminuiria sua relevância. A outra é o fato de que é bastante complexo atingir uma delimitação exata do alvo em diversas aplicações, assim como no Kinesis. Devido a isto, o alvo selecionado para o rastreamento provavelmente conterá alguma característica do fundo.

Sendo assim, o uso inapropriado de informações provenientes do fundo da imagem pode influenciar negativamente no processo de estimativa de localização e na escolha da escala do alvo. Desta forma, a opção para o sistema Kinesis foi

representar as características do fundo, usando-as para seleccionar apenas as partes salientes da representação do modelo do alvo e de seus candidatos.

#### 4.7 Minimização da Distância

Como visto anteriormente, no sistema Kinesis, a localização mais provável do alvo no *frame* atual é baseada na minimização da distância, o que corresponde à maximização do coeficiente de Bhattacharyya  $\hat{\rho}(y)$ . A busca pela nova localização do alvo é iniciada em  $\hat{y}_0$ , o qual é estimado do alvo no *frame* anterior. Assim, a probabilidade das cores do candidato do alvo  $\{\hat{p}_u(\hat{y}_0)\}_{u=1\dots m}$  na localização  $\hat{y}_0$  deve ser computada primeiramente. De acordo com Ferrugem et al. (2004), usando a expansão de Taylor nos valores  $\hat{p}_u(\hat{y}_0)$ , o coeficiente de Bhattacharyya pode ser aproximado (após algumas manipulações) por:

$$[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} + \frac{1}{2} \sum_{u=1}^m \hat{p}_u(y) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \quad (22)$$

Introduzindo (17) em (22), obtém-se:

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} + \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k \left( \left\| \frac{y - x_i}{h} \right\|^2 \right) \quad (23)$$

onde:

$$w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \delta[b(x_i) - u] \quad (24)$$

Desta forma, para se minimizar a distância, o segundo termo na equação (23) deve ser maximizado, visto que o primeiro é independente de  $y$ . O segundo termo representa a estimativa da densidade computada com o perfil de *kernel*  $k$  em  $y$  no *frame* corrente, com os dados tendo pesos atribuídos conforme  $w_i$ .

## 4.8 Extração de Métricas

Segundo Schnack (2005), a principal funcionalidade do sistema Kinesis não é apenas efetuar o rastreamento de jogadores de alto nível em esportes coletivos, mas sim, a partir do comportamento do alvo, extrair certas métricas que venham a auxiliar o treinamento dos atletas. Seriam elas: distância percorrida, velocidade e aceleração do jogador.

Para realizar esta extração, se faz necessário converter as coordenadas da imagem para uma representatividade no mundo real. Ou seja, converter as coordenadas de *pixels* para uma unidade de distância como, por exemplo, o metro. Assim sendo, o usuário deve selecionar a quantidade de *pixels* que correspondem a um metro, em ambos os eixos ( $x$ ) e ( $y$ ).

Ao longo do processo de rastreamento, o centro do modelo do alvo em cada *frame* é armazenado, formando a seqüência de valores  $C = \{c_i\}_{i=1\dots n}$ , onde  $n$  é o número total de *frames*. Então, a métrica da distância total percorrida é obtida por:

$$d = \sum_{i=2}^n c_i - c_{i-1} \quad (25)$$

A diferença entre  $c_i$  e  $c_{i-1}$  representa quanto o alvo se locomoveu do frame  $i-1$  para o frame  $i$ . Dado que o rastreamento é efetuado em um vídeo 2D, o cálculo da distância entre os dois pontos é efetuado por:

$$d_i = \sqrt{(c_i \cdot x - c_{i-1} \cdot x)^2 + (c_i \cdot y - c_{i-1} \cdot y)^2} \quad (26)$$

Entretanto, essa equação (anterior) não leva em conta a representação da distância em *pixel* no mundo real. Dessa forma, o cálculo da distância real é dado por:

$$d_i = \sqrt{[p_x(c_i \cdot x - c_{i-1} \cdot x)]^2 + [p_y(c_i \cdot y - c_{i-1} \cdot y)]^2} \quad (27)$$

onde  $p_x$  e  $p_y$  são o número de *pixels* de  $x$  e  $y$ , respectivamente, que representam a unidade de medida no mundo real.

A velocidade é dada pela distância percorrida em relação ao tempo. Assim sendo, da mesma forma que a distância é obtida pela diferença na posição entre

duas amostras consecutivas (antepenúltima acima), a velocidade é calculada através da diferença da distância em relação ao tempo entre elas:

$$v_i = C_m C_t F (d_i - d_{i-1}) \quad (28)$$

sendo  $C_m$  uma constante para converter a unidade de medida para quilômetros,  $C_t$  o fator de conversão de segundos para hora e  $F$  a quantidade de *frames* por segundo no vídeo analisado.

A aceleração é dada pela variação da velocidade:

$$a_i = (v_i - v_{i-1}) \quad (29)$$

Em todas as métricas (distância, velocidade e aceleração) é assumido que no estado inicial o alvo está em repouso. Assim sendo, a seqüência da distância, da velocidade e da aceleração é inicializada por 0.

Com base na métrica de velocidade, é possível classificar o movimento do alvo, ou seja, categorias são criadas para efetuar a discretização da velocidade. Uma possível classificação é mostrada na Tab. 2.

Tabela 2. Categorias de movimento a partir da velocidade.

<b>Faixa de Velocidade</b>	<b>Movimento</b>
Velocidade abaixo de 4 km/h	caminhada
Velocidade entre 4 km/h e 8 km/h	trote
Velocidade entre 8 km/h 12 km/h	corrida
Velocidade acima de 12 km/h	<i>sprint</i>

Fonte: SCHNACK, M. 2005.

A partir da informação da localização do alvo em todos os frames, a movimentação do jogador é recriada em um modelo mais simples. Esse modelo visa melhorar a análise da movimentação do alvo após a partida e a facilitar a transmissão dela para outros dispositivos como, por exemplo, aparelhos de telefonia móvel.



## 4.9 Algoritmo

Schnack (2005) descreveu o funcionamento do algoritmo de *mean shift* conforme empregado no sistema Kinesis, nos seguintes passos:

Dado a distribuição  $\{\hat{q}_u\}_{u=1\dots m}$  do modelo do alvo e a localização estimada do alvo  $\hat{y}_0$  no frame anterior:

1. Inicializar a localização do alvo no frame corrente com  $\hat{y}_0$ , computar a distribuição  $\{\hat{p}_u(\hat{y}_0)\}_{u=1\dots m}$  e calcular

$$\rho[\hat{p}(\hat{y}_0), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} \quad (30)$$

2. Calcular os pesos  $\{w_i\}_{i=1\dots n_h}$  de acordo com

$$w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \delta[b(x_i) - u] \quad (31)$$

3. Baseado no vetor de *mean shift*, a nova posição do alvo é estimada

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)} \quad (32)$$

4. Computar  $\{\hat{p}_u(\hat{y}_1)\}_{u=1\dots m}$  e avaliar

$$\rho[\hat{p}(\hat{y}_1), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_1) \hat{q}_u} \quad (33)$$

5. Enquanto  $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$  fazer  $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_0 + \hat{y}_1)$ .
6. Se  $\|\hat{y}_1 - \hat{y}_0\| < \epsilon$  parar, Senão, atribuir  $\hat{y}_0 \leftarrow \hat{y}_1$  e ir para o passo 2. Onde *parar* representa o critério de parada do algoritmo.

Schnack (2005) ainda afirma que o *threshold* do critério de parada utilizado no passo 6 consiste em assegurar que os vetores representados por  $\hat{y}_0$  e  $\hat{y}_1$  estejam na mesma coordenada de *pixel*. Caso exista a necessidade de executar o procedimento em tempo real, é interessante utilizar outro critério de parada, que

limite o número de iterações do *mean shift* para  $N_{max}$ , usualmente igual a 20. Na prática, a média do número de iterações é muito menor, em torno de 4.

A implementação do algoritmo proposto pode ser ainda mais simples que a apresentada acima. O papel do passo 5 é apenas evitar potenciais problemas numéricos decorrentes da aproximação do coeficiente de Bhattacharyya. Entretanto, experimentos práticos demonstraram que o coeficiente computado em  $\hat{y}_1$  não aumenta em apenas 0,1% dos casos. Assim sendo, o passo 5 não é utilizado na prática e, como resultado, o coeficiente não precisa ser calculado nos passos 1 e 4.

No algoritmo utilizado na prática, apenas existe o cálculo do passo 2, a estimativa da nova localização do passo 3 e o teste de parada do passo 6. O coeficiente de Bhattacharyya é apenas calculado após a finalização desse algoritmo, para se medir a similaridade do modelo do alvo com o candidato escolhido.

#### **4.10 Situações Críticas**

Alguns problemas foram encontrados no rastreamento do sistema Kinesis, durante a fase de testes da primeira versão. Três destes problemas são a oclusão quase total, a oclusão total e a mudança ou deformação acentuada do alvo a ser rastreado. Visando uma melhora de desempenho no rastreamento do Kinesis, ou até mesmo a solução de alguns destes problemas citados, este trabalho propõe o estudo e a aplicação, como apoio ao *mean shift*, do filtro de Kalman, assunto principal do próximo capítulo.

## 5 O Filtro de Kalman

Publicado inicialmente por Kalman, R. (1960), e recebendo o nome de seu criador, o filtro de Kalman (FK) surgiu como uma nova alternativa para filtragem linear e problemas de predição. Sua primeira descrição tratava-se de uma versão em tempo contínuo, contando com a colaboração de Richard S. Bucy e, portanto, também conhecida como filtro de Kalman-Bucy. Baseado no modelo de Wiener (1949 apud KALMAN, 1960), este novo filtro buscava realizar da mesma forma a especificação de sistemas lineares, porém com o diferencial de comportar predição, separação ou detecção de um sinal de ruído aleatório.

De acordo com Leandro (2006), ambos os filtros tinham motivação centrada em fins militares. O filtro de Wiener tinha como objetivo posicionar um canhão, durante a Segunda Guerra Mundial, a fim de que o projétil atingisse uma aeronave, com o menor erro possível, caracterizando uma série temporal estacionária. Já o filtro de Kalman previa a trajetória de mísseis balísticos cujas trajetórias, nas fases de lançamento e chegada, têm comportamento não-estacionário.

Neste trabalho, o FK é aplicado junto ao sistema Kinesis como apoio ao rastreamento realizado pelo algoritmo de *mean shift*. Com isto, objetiva-se uma melhora de desempenho nas situações de oclusão ou deformação do alvo.

Este capítulo introduz o filtro de Kalman padrão, empregado em sistemas lineares, além de outras extensões desta ferramenta para sistemas não-lineares. Também são apresentados algoritmos, aplicações e exemplos destas técnicas.

### 5.1 O Processo de Estimação Utilizando Filtros

Filtragem é a estimação do estado atual de um sistema dinâmico. A escolha da palavra “filtro” para esta tarefa refere-se ao processo que visa uma estimação

ótima a partir de porções de dados ruidosos, eliminando estes ruídos de forma análoga a um filtro de líquidos, por exemplo (BAR-SHALOM, 2001).

Um estimador ótimo é um algoritmo computacional que processa medições ou observações fornecendo estimativas de uma variável de interesse, otimizando um critério particular. Sendo assim, o processo de estimação pode ser visto como um esquema para extração de informações e aprimoramento, pois, com base nas medições, objetiva-se maximizar o conhecimento a respeito de um parâmetro, estado, sinal, imagem ou afins.

Bar-Shalom (2001) propõe uma classificação das variáveis a serem estimadas em duas categorias. São elas:

- Um parâmetro - quantidade variante no tempo (um escalar, um vetor, uma matriz);
- Um estado de um sistema dinâmico (normalmente um vetor), que evolui ao longo do tempo, de acordo com uma equação estocástica.

Desta forma, duas classes de estimadores são derivadas: os estimadores de parâmetro e os estimadores de estado. A Fig. 12 apresenta um diagrama em blocos que ilustra o processo de estimação de estado. Os dois primeiros blocos são considerados “caixas-pretas”, pois não se tem acesso aos dados internos a eles. As únicas variáveis às quais o estimador tem acesso são as medições, as quais são afetadas por fontes de erros na forma de ruídos. O estimador utiliza conhecimento sobre a evolução da variável, o modelo de medição, a caracterização probabilística dos diversos fatores aleatórios (perturbações) e informações prévias.

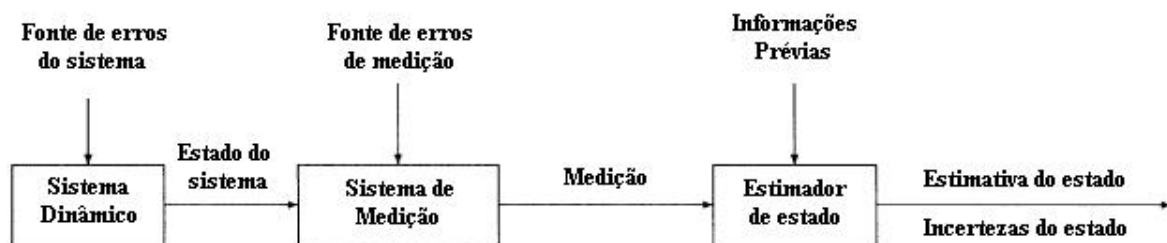


Figura 12: Estimação de estado: extração de informação e aprimoramento.

Fonte: BAR-SHALOM, Y., 2001.

## 5.2 Ciclo de Operação e Recursividade

O filtro de Kalman possui basicamente dois blocos de operação, sendo eles as fases de previsão e atualização do sistema. A primeira envolve a previsão do estado e da covariância. Já a segunda envolve o cálculo do ganho de Kalman, e atualização do estado e da covariância.

De acordo com Welch e Bishop (1999), a atualização de tempo projeta o estado atual para o instante posterior, enquanto a atualização da medição ajusta a estimativa projetada por uma medida real naquele instante de tempo. A Fig. 13 ilustra o ciclo de operação do filtro de Kalman.

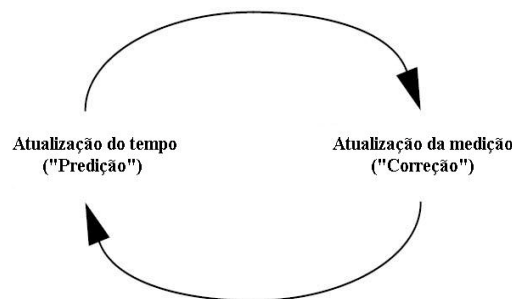


Figura 13: Ciclo de operação do filtro de Kalman discreto.

Fonte: WELCH, G; BISHOP, G., 1999.

De forma semelhante, porém mais elaborada (Fig.14), Castellanos (2004) ilustra a estrutura do filtro de Kalman, na qual as entradas da planta física são também entradas do modelo de previsão, e a saída do modelo de previsão é comparada com a saída da planta física, admitindo a variável a ser estimada como a velocidade de um alvo qualquer, apenas para fins de exemplificação. O erro calculado desta comparação, junto com os estados estimados, fornece a informação ao bloco de correção do filtro de Kalman que permite reduzir o erro de previsão de estado.

Segundo Velhote (2005), em um caso prático de aplicação do FK é necessário:

- Definir o vetor de variáveis de estado  $x$ ;
- Identificar a matriz  $A$  que representa a dinâmica de transição do estado;

- Identificar a matriz  $H$  que representa a relação entre as variáveis mensuráveis (normalmente em número inferior) e as variáveis de estado do sistema;
- Avaliar as incertezas no modelo do sistema e dos instrumentos de medição, representando-as como covariâncias das suas respectivas variáveis;
- Conhecer as condições iniciais das variáveis de estado.

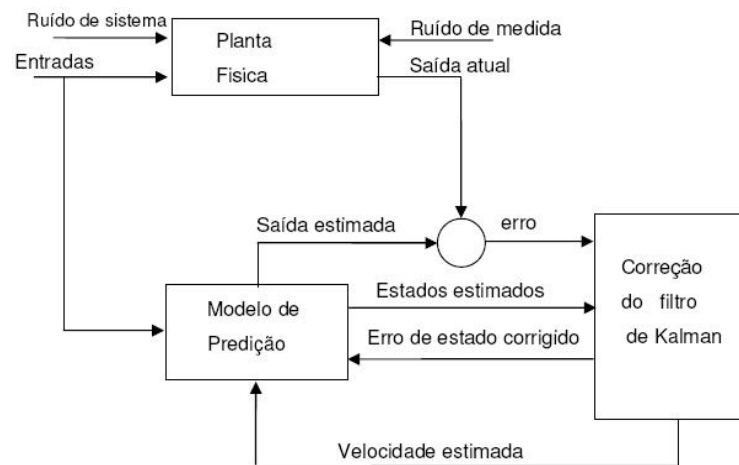


Figura 14: Estrutura detalhada do ciclo do filtro de Kalman.

Fonte: CASTELLANOS, J. A. G., 2004.

Com isto, pode-se afirmar que o algoritmo de funcionamento do filtro de Kalman é recursivo, devido à utilização de diferentes períodos de tempo ao longo de cada estimativa que compõe o processo como um todo. Welch e Bishop (2001) ilustram o processo completo de operação do FK, incluindo as equações básicas, conforme mostrado na Fig. 15. Estas equações pertencem ao filtro de Kalman padrão e serão apresentadas, com mais detalhes, mais adiante neste capítulo.



Figura 15: Ciclo de operação do filtro de Kalman discreto.

Fonte: WELCH, G; BISHOP, G., 1999.

As equações de atualização do tempo são responsáveis pelo avanço das variáveis de estado e das covariâncias no tempo para se obter, desta forma, as estimativas anteriores (*a priori*) para o próximo instante, sendo, por isto, denominadas de equações de previsão. Já as equações de atualização das medições são responsáveis pela retroalimentação, ou seja, incorporam uma nova informação da variável observável nas estimativas anteriores para obter um ganho (ou melhoria) na estimação posterior, sendo denominadas de equações de correção.

### 5.3 Sistemas Lineares e o Filtro de Kalman Padrão

Segundo Simon (2006), diversos processos podem ser descritos por sistemas que utilizam relação estado-estado. Isto inclui problemas de engenharia, economia, física, química, biologia, e diversas outras áreas. Se for possível derivar um modelo matemático para um processo, então também será possível utilizar ferramentas matemáticas para controlar o processo e obter informações sobre o mesmo. Se o estado do sistema no período de tempo atual é conhecido, bem como todas as entradas atuais e futuras, então se podem deduzir os valores de todas as saídas futuras do sistema.

Modelos estado-estado podem ser basicamente divididos em modelos lineares e não-lineares. Entretanto, a grande maioria dos processos reais é não-

linear e as ferramentas matemáticas designadas para estimação e controle são muito mais acessíveis e fáceis de entender para sistemas lineares. Por este motivo, sistemas não-lineares são freqüentemente aproximados a sistemas lineares. Com isto, torna-se possível a utilização das ferramentas matemáticas citadas anteriormente.

Schutter et al. (1999) define o filtro de Kalman como um estimador linear, baseado em modelo, estocástico, recursivo, ponderado e baseado em mínimos quadrados. É linear e baseado em modelo, porque utiliza como base um modelo de sistema cujas equações de entrada e de saída são todas lineares. Estocástico, porque a confiança das informações é expressa em termos de distribuições de probabilidade. A distribuição Gaussiana é utilizada pelo FK para a medição e a estimação do estado. É recursivo, pois nos casos em que as informações se tornam disponíveis de maneira incremental, uma formulação recursiva do processo de estimação se faz necessária. É baseado em mínimos quadrados porque casualmente a estimativa ultrapassa os limites de determinação, impedindo uma consistência perfeita com todas as informações coletadas a respeito do sistema (medições). O ruído branco Gaussiano e de média zero, é uma das importantes fontes de erro. Então, o FK provê uma estimação de forma a tentar minimizar as inconsistências com todas as partes da informação em um senso mínimo quadrático. Com respeito a isto, o FK é classificado como um estimador ótimo. É considerado ponderado devido ao fato de quando os cálculos dos valores mínimos quadrados são minimizados, inconsistências com diferentes partes da informação têm pesos atribuídos a si, com uma medição de certeza desta informação, de forma que informações imprecisas são atribuídas de pesos baixos, enquanto às mais precisas atribuem-se pesos bastante elevados.

A função principal é estimar o estado de um sistema, ou parte dele, baseado em conhecimento (medição) das entradas e saídas deste sistema. Grewal e Andrews (2001) apresentam alguns conceitos fundamentais para o FK, conforme mostrados na Fig. 16.





Figura 16: Conceitos Fundamentais para o filtro de Kalman.

Fonte: SIMON, D., 2006.

Em suma, pode-se afirmar que o FK é um conjunto de equações matemáticas que constitui um processo recursivo eficiente de estimação, uma vez que o erro quadrático é minimizado. Através da observação da variável denominada “variável de observação” outra variável, não observável, denominada “variável de estado” pode ser estimada eficientemente. Podem ser estimados os estados passados, o estado presente e, mesmo, previstos os estados futuros.

### 5.3.1 Exemplo Intuitivo

Como comentado anteriormente, o filtro de Kalman é uma ferramenta para fusão de dados estocásticos. Segundo Schutter et al. (1999), isto pode ser ilustrado com o exemplo a seguir. Considere-se uma quantidade ( $x$ ), medida duas vezes com equipamentos iguais ou diferentes. As duas medições são denotadas como  $x_1$  e  $x_2$ , e caracterizadas por distribuições de probabilidade Gaussianas, com médias  $\bar{x}_1$  e  $\bar{x}_2$  e desvios padrão  $\sigma_1$  e  $\sigma_2$ :

$$p(x_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp \left[ -\left( \frac{x_i - \bar{x}_i}{\sigma_i} \right)^2 \right] \quad (34)$$

com  $i = (1,2)$ .

Os desvios padrão são interpretados como uma medida de incerteza. Estas medidas são combinadas para fornecer uma estimativa da quantidade ( $x$ ):

$$\hat{x} = wx_1 + (1 - w)x_2 \quad (35)$$

onde  $(\hat{\cdot})$  é a estimativa da média, e  $w$  é um peso ainda a ser determinado.

Pelo fato de  $x_1$  e  $x_2$  possuírem distribuições Gaussianas,  $\hat{x}$  tem também distribuição Gaussiana com desvio padrão  $\hat{\sigma}$ , dado por:

$$\hat{\sigma}^2 = w^2 \sigma_1^2 + (1 - w)^2 \sigma_2^2 \quad (36)$$

O peso que minimiza essa incerteza, expressado por  $\hat{\sigma}$ , é encontrado como (admitindo  $\frac{\partial}{\partial w} = 0$ ):

$$w_{opt} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \quad (37)$$

Disto, as equações (35) e (36) passam a ser:

$$\hat{x} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} x_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} x_2 \quad (38)$$

e:

$$\sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

A estimativa  $\hat{x}$  minimiza a soma das distâncias até  $x_1$  e  $x_2$ , com pesos atribuídos de acordo com os respectivos desvios padrão:

$$\hat{x} = \arg \min_x \left[ \left( \frac{x_1 - x}{\sigma_1} \right)^2 + \left( \frac{x_2 - x}{\sigma_2} \right)^2 \right] \quad (39)$$

admitindo novamente  $\frac{\partial}{\partial w} = 0$ .

Supondo que as duas medições sejam disponibilizadas sequencialmente. Ao 1º passo de tempo, a medição  $x_1$  se torna disponível. Desde que esta seja a única informação, a estimativa de estado e sua variância são  $\hat{x}_1 = x_1$  e  $\hat{\sigma}_1^2 = \sigma_1^2$  (o índice subscrito se refere ao passo de tempo atual). Então, no 2º passo, a medição  $x_2$  passa a estar disponível, e a estimação é novamente realizada, como em (38), porém reescrita em uma forma recursiva:

$$\hat{x} = \hat{x}_1 + \frac{\hat{\sigma}_1^2}{\hat{\sigma}_1^2 + \sigma_2^2} (x_2 - \hat{x}_1) \quad (40)$$

e:

$$\hat{\sigma}_2^2 = \left(1 - \frac{\hat{\sigma}_1^2}{\hat{\sigma}_1^2 + \sigma_2^2}\right) \hat{\sigma}_1^2 \quad (41)$$

com  $\hat{x}_2 - x_1$  representando a nova informação recebida no 2º passo, e chamada de inovação. Além disso,

$$K = \frac{\hat{\sigma}_1^2}{\hat{\sigma}_1^2 + \sigma_2^2} \quad (42)$$

é chamada de ganho de atualização. Assim, (40) e (41) passam a ser:

$$\hat{x}_2 = \hat{x}_1 + K(x_2 - \hat{x}_1) \quad (43)$$

e

$$\hat{\sigma}_2^2 = (1 - K)\hat{\sigma}_1^2 \quad (44)$$

lembrando novamente que os índices subscritos se referem ao passo de tempo em que a medição foi tomada ou a estimativa foi computada.

### 5.3.2 Sistemas Estáticos

De acordo com Schutter et al. (1999), um sistema linear estático possui um vetor de estado constante  $x$ , de dimensão  $n$  e uma equação de saída (medição) linear:

$$z = Hx + \rho_m \quad (45)$$

onde  $z$  é um vetor  $l$ -dimensional de medição.  $\rho_m$  representa uma incerteza de média zero ( $\bar{\rho}_m = 0$ ) na medição e matriz de covariância  $R = E[(\rho_m - \bar{\rho}_m) (\rho_m - \bar{\rho}_m)^T]$ , onde  $E[.]$  corresponde ao valor esperado. Sendo assim, comparando com o exemplo da seção anterior, neste caso não é possível medir as variáveis de estado diretamente, mas apenas com combinações lineares, conforme expressado pela equação (45).

Suponhamos que  $N$  medições  $z_k$  estejam disponíveis. Cada medição pode ter sua própria relação de medida  $H_k$  e matriz de covariância  $R_k$ . Então, a estimativa

de estado  $\hat{x}$  que minimiza as distâncias mínimas quadradas ponderadas entre as medições  $z_k$  e a medição esperada,  $\hat{z}_k = H_k \hat{x}$ , é dada por:

$$\hat{x} = \arg \min_x \sum_{k=1}^N (z_k - H_k x)^T R_k^{-1} (z_k - H_k x) \quad (46)$$

observando que na seção anterior, o espaço de medição coincide com o espaço de estado. Portanto, a equação (39) corresponde à equação (46), com a substituição de  $z_k$  por  $x_k$ , e admitindo  $H_k = 1$ .

Informação inicial a respeito do vetor de estado e sua matriz de covariância,  $\hat{x}_0$  e  $\hat{P}_0$  podem ser adicionadas à equação (46), tratando isto como a primeira medição no passo de tempo 0, ou seja,  $z_0 = \hat{x}_0$ ,  $H_0 = I$  e  $R_0 = \hat{P}_0$ :

$$\hat{x} = \arg \min_x \left[ (\hat{x}_0 - x)^T \hat{P}_0^{-1} (\hat{x}_0 - x) + \sum_{k=1}^N (z_k - H_k x)^T R_k^{-1} (z_k - H_k x) \right] \quad (47)$$

Se as medições se tornam disponíveis seqüencialmente, e a cada instante  $k$  é possível adicionar uma nova informação de forma a obter uma nova estimativa  $\hat{x}_k$  e uma correspondente matriz de covariância  $\hat{P}_k$ , então a equação (47) pode ser reescrita recursivamente:

$$\hat{x}_k = \arg \min_x \left[ (\hat{x}_{k-1} - x)^T \hat{P}_{k-1}^{-1} (\hat{x}_{k-1} - x) + (z_k - H_k x)^T R_k^{-1} (z_k - H_k x) \right] \quad (48)$$

A solução deste problema de minimização é chamada de filtro de Kalman Estático. Este computa uma nova estimativa  $\hat{x}_k$  e sua matriz de covariância  $\hat{P}_k$  a partir de  $\hat{x}_{k-1}$ ,  $\hat{P}_{k-1}$ ,  $z_k$  e  $R_k$ .

### 5.3.3 Sistemas Dinâmicos

Ao definir um sistema dinâmico, Schutter et al. (1999), propõe inicialmente a suposição do estado de um sistema evoluindo de acordo com a seguinte equação linear de estado:

$$x_k = Ax_{k-1} + Bu_{k-1} + \rho_p \quad (49)$$

onde  $A$  é a matriz de estado,  $B$  a matriz de entrada,  $u_{k-1}$  é o vetor de entrada de dimensão  $r$  no passo de tempo  $k - 1$ , e  $\rho_p$  representa a incerteza do processo, com matriz de covariância  $Q$  (ou  $Q_{k-1}$  se a incerteza do processo for variante no tempo).

O vetor de estado evolui entre os instantes de tempo  $k - 1$  e  $k$ . Portanto, o primeiro termo da equação (48) precisa ser alterado, baseado em uma predição desta evolução. A predição do estado  $\tilde{x}_k$  é derivada da equação (49):

$$\tilde{x}_k = A\hat{x}_{k-1} + Bu_{k-1} \quad (50)$$

Se a entrada  $u_k$  for perfeitamente conhecida, então o erro de predição é encontrado a partir da subtração das equações (50) e (51):

$$\tilde{x}_k - x_k = A(\tilde{x}_{k-1} - x_{k-1}) - \rho_p \quad (51)$$

Desta forma, a matriz de covariância de  $\tilde{x}_k$  é:

$$\tilde{P}_k = A\hat{P}_{k-1}A^T + Q_{k-1} \quad (52)$$

Substituindo  $\tilde{x}_k$  e  $\tilde{P}_k$  por  $\hat{x}_{k-1}$  e  $\hat{P}_{k-1}$  na equação (48), temos:

$$\hat{x}_k = \arg \min_x [(\tilde{x}_k - x)^T \tilde{P}_k^{-1} (\tilde{x}_k - x) + (z_k - H_k x)^T R_k^{-1} (z_k - H_k x)] \quad (53)$$

O mínimo é novamente encontrado admitindo que a derivada parcial relativa ao cálculo de  $x$  seja igual a 0. Disto, tem-se:

$$\hat{x}_k = \tilde{x}_k + K_k(z_k - H_k \tilde{x}_k) \quad (54)$$

e:

$$\hat{P}_k = (I - K_k H_k) \tilde{P}_k \quad (55)$$

com:

$$K_k = \tilde{P}_k H_k^T S_k^{-1} \quad (56)$$

e:

$$S_k = R_k + H_k \tilde{P}_k H_k^T \quad (57)$$

onde  $K$  é chamado de ganho de Kalman, e  $S$  é a matriz de covariância.

As equações (51), (52) e (54)-(57) constituem o que se chama de filtro de Kalman Dinâmico.  $v_k = z_k - H_k \tilde{x}_k$  é a nova informação no instante de tempo  $k$  e é

chamada de inovação, que representa a diferença entre os valores reais e os medidos pela predição do filtro, também conhecido como filtro de Kalman Padrão.

#### 5.4 Sistemas Não-Lineares

De acordo com Simon (2006), o filtro de Kalman é um filtro linear, e por isto, obviamente, só pode ser aplicado a um sistema linear. Infelizmente, a grande maioria dos sistemas físicos reais é não-linear. Muitos sistemas são próximos o bastante de um sistema linear e, por isso, a aplicação do filtro de Kalman pode gerar resultados satisfatórios. Porém, diversos outros sistemas só se comportam de forma linear quando submetidos a uma pequena faixa de operação, tendo comportamento não-linear fora desta faixa. Nestes casos, o filtro de Kalman padrão não mais alcançará bons resultados, sendo necessária então a exploração de filtros não-lineares.

A filtragem não-linear é demasiada complexa e de difícil entendimento em relação à filtragem linear. Entretanto, alguns métodos de estimação não-linear têm sido mais difundidos. Estes métodos incluem extensões não-lineares do filtro de Kalman, filtros *Unscented* e filtros de partículas. Neste trabalho são apresentadas as extensões do FK mais conhecidas e utilizadas para os problemas em questão.

Segundo Grewal e Andrews (2001), alguns aspectos devem ser levados em conta ao se utilizar o FK em sistemas não-lineares. Seriam eles:

- Muitos problemas de estimação de interesse prático são não-lineares, mas podem ser aproximados dos lineares através de pequenas perturbações nos valores das variáveis de estado;
- Métodos da teoria da estimação linear podem ser aplicados a alguns problemas não-lineares por uma aproximação dos efeitos das pequenas perturbações no estado do sistema não-linear através de um valor nominal;
- Para alguns problemas, os valores nominais das variáveis de estado são conhecidos anteriormente. Estes incluem aplicações de guia e controle, para os quais a performance operacional depende de se manter próxima à trajetória ótima. Para estas aplicações, o problema da estimação pode ser até mesmo efetivamente linearizado em relação à sua trajetória

nominal, e o ganho de Kalman pode ser pré-computado pra aliviar a carga computacional do tempo-real;

- A trajetória nominal pode ser alternativamente definida em um determinado instante de tempo como a melhor estimativa atual da trajetória. Esta aproximação é conhecida como filtro de Kalman estendido, o qual é apresentado mais adiante neste capítulo. Sua vantagem é a de que as perturbações incluem apenas os erros de estimação de estado, os quais são geralmente menores do que as perturbações de qualquer trajetória nominal predefinida e, portanto, melhor condicionadas para uma aproximação linear. A maior desvantagem é a adição do custo computacional de tempo-real da linearização acerca de uma trajetória imprevisível, para a qual o ganho de Kalman não pode ser computado anteriormente;
- Extensões do modelo linear que incluem termos quadráticos fornecem filtros ótimos de grande aplicabilidade, porém aumentam a complexidade computacional.

#### 5.4.1 Linearização pela Expansão em Série de Taylor

Para Simon (2006), a chave para filtragem não-linear utilizando filtro de Kalman é expandir os termos não-lineares da equação do sistema através da expansão em série de Taylor, em torno de um ponto  $\bar{x}$ . A expansão em série de Taylor de uma função não-linear pode ser escrita como:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(\bar{x})\Delta x^n}{n!} \quad (58)$$

$$= f(\bar{x}) + f'(\bar{x})\Delta x + f''(\bar{x})\Delta x^2/2 + \dots$$

onde  $\Delta x = x - \bar{x}$  e  $f^{(n)}(\bar{x})$  é a  $n$ -ésima derivada de  $f(x)$ , avaliada em  $x = \bar{x}$ .

Para facilitar a compreensão, segue um exemplo. Supondo que se queira expandir  $f(x) = \cos(x)$  pela expansão em série de Taylor, em torno do ponto  $\bar{x} = 0$ . Cabe lembrar que a derivada de  $\cos(x)$  é  $-\sin(x)$ , e a derivada de  $\sin(x)$  é  $\cos(x)$ . A partir disto, pode-se escrever a expansão em série de Taylor de  $\cos(x)$  como:

$$\cos(x) = \cos(\bar{x}) - \text{sen}(\bar{x})\Delta x - \cos(\bar{x})\Delta x^2/2 + \dots \quad (59)$$

A partir da expansão de  $\cos(x)$  em torno do ponto nominal  $x = 0$ , pode-se notar que  $\bar{x} = 0$  e  $\Delta x = x - \bar{x} = x$ . Assim, a expansão em série de Taylor passa a ser descrita por:

$$\begin{aligned} \cos(x) &= \cos(0) - \text{sen}(0)\Delta x - \cos(0)\Delta x^2/2 + \dots \\ &= 1 - \frac{x^2}{2} + \dots \end{aligned} \quad (60)$$

Se utilizarmos a expansão em série de Taylor de segunda ordem para  $\cos(x)$ , então podemos afirmar que  $\cos(x)$  é aproximadamente igual a  $1 - x^2/2$  (é chamada de “segunda ordem” porque trabalha com a segunda potência de  $x$ ). Em outras palavras, podemos ignorar os outros termos da série de Taylor, porque envolvem potências maiores de  $x$ , as quais são divididas por fatoriais crescentes, o que os torna insignificantes comparados aos termos de ordem mais baixa. A Tab. 3 mostra  $\cos(x)$  e sua expansão em série de Taylor de segunda ordem para valores variáveis de  $x$ . Pode-se notar que quanto menor for o valor de  $x$ , ou seja, quanto mais próximo ele for do ponto nominal  $\bar{x} = 0$ , melhor é a aproximação fornecida pela expansão em série de Taylor em relação ao valor real de  $\cos(x)$ .

Tabela 3. Expansão em série de Taylor de  $\cos(x)$  para valores variáveis de  $x$ .

$x$	$\cos(x)$	$1 - x^2/2$
0	1	1
0,25	0,969	0,969
0,5	0,878	0,875
0,75	0,732	0,719
1,00	0,540	0,500
1,25	0,315	0,219
1,50	0,071	-0,125

Fonte: SIMON, D. 2006.

Linearizar uma função significa expandir a mesma em uma série de Taylor de primeira ordem, em torno de um ponto de expansão. Em outras palavras, a



expansão em série de Taylor de primeira ordem para a função  $f(x)$  é equivalente a  $f(x) = f(\bar{x}) + f'(\bar{x})\Delta x$ .

#### 5.4.2 O Filtro de Kalman Linearizado

Segundo Simon (2006), a idéia do filtro de Kalman linearizado é iniciar com um sistema não-linear, e então encontrar um sistema linear cujos estados representem os desvios de uma trajetória nominal do sistema não-linear. Assim, o filtro de Kalman pode ser usado para estimar tais desvios, fornecendo também, de forma indireta, a estimação dos estados do sistema não-linear.

Um modelo básico para um sistema não-linear pode ser descrito como:

- Equação de estado:  $x_{k+1} = f(x_k, u_k) + w_k$
- Equação de saída:  $z_k = h(x_k) + v_k$

onde a equação de estado  $f(\cdot)$  e a equação de medição  $h(\cdot)$  são equações não-lineares.

Suponhamos então o seguinte exemplo, proposto por Simon (2006):

- Equação de estado:  $x_{k+1} = x_k^2 + \cos x_k + u_k + w_k$
- Equação de saída:  $z_k = 1/x_k + v_k$

A equação de estado é composta por dois termos não-lineares, um termo quadrado e um cosseno. A equação de saída também é não-linear, pois contém o termo  $1/x_k$ . Se ambas as equações de estado e de saída são não-lineares, então o sistema como um todo é considerado não-linear.

No filtro de Kalman linearizado, se utiliza a série de Taylor de primeira ordem para expandir a equação de estado e saída de um sistema, em torno de um estado nominal, conforme explicado anteriormente. Este estado nominal é uma função de tempo, portanto pode ser referenciado, às vezes, como uma trajetória. Esta é baseada em uma estimativa a respeito do que o comportamento do sistema representa. Como exemplo pode-se citar o sistema Kinesis, onde as equações do sistema representam a dinâmica da movimentação de um atleta e, portanto, o estado nominal seria a trajetória percorrida por este atleta. Assim, a trajetória estimada do atleta poderia ser diferente da trajetória nominal, devido a não-linearidade do movimento do atleta, distúrbios, erros em modelagem, ou outros erros inesperados. Porém, espera-se que a trajetória estimada seja próxima à nominal, no

caso da linearização pela expansão em série de Taylor ser eficaz o suficiente. As linearizações pela série de Taylor para as equações de estado e saída do sistema não-linear podem ser representadas por:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + w_k \\ &\approx f(\bar{x}_k, u_k) + f'(\bar{x}_k, u_k)\Delta x_k + w_k \end{aligned} \quad (61)$$

e

$$\begin{aligned} z_k &= h(x_k) + v_k \\ &\approx h(\bar{x}_k) + h'(\bar{x}_k)\Delta x_k + v_k \end{aligned} \quad (62)$$

Visto isso, os desvios para a trajetória nominal podem ser escritos como:

$$\begin{aligned} \Delta x_{k+1} &= x_{k+1} - \bar{x}_{k+1} \\ &= x_{k+1} - f(\bar{x}_k, u_k) \end{aligned} \quad (63)$$

e

$$\begin{aligned} \Delta z_k &= z_k - \bar{z}_k \\ &= z_k - h(\bar{x}_k) \end{aligned} \quad (64)$$

Combinando estas equações acima com as suas anteriores, tem-se:

$$\Delta x_{k+1} = f'(\bar{x}_k, u_k)\Delta x_k + w_k \quad (65)$$

e

$$\Delta z_k = h'(\bar{x}_k)\Delta x_k + v_k \quad (66)$$

Desta forma, foram obtidas equações de estado e saída que são funções lineares de  $\Delta x$  e  $\Delta z$ . Isto significa que é possível usar o filtro de Kalman padrão para estimar  $\Delta x$ .

Existem dois pontos importantes a serem lembrados ao utilizar o filtro de Kalman linearizado, segundo Simon (2006):

- Após o FK ser usado para estimar  $\Delta x$ , é necessário somar a estimativa de  $\Delta x$  ao estado nominal  $\bar{x}$ , de forma a obter uma estimativa do estado  $x$ . Isto porque  $\Delta x = x - \bar{x}$ .
- Se o estado real  $x$  ficar muito distante do estado nominal  $\bar{x}$ , o filtro de Kalman linearizado não terá bons resultados. Isto ocorre porque a aproximação pela série de Taylor interrompe se  $x$  torna-se muito distante de  $\bar{x}$ .

Visto isto, o filtro de Kalman linearizado pode ser resumido pelo seguinte algoritmo:

1. As equações do sistema são dadas por:

$$\text{Equação de estado: } x_{k+1} = f(x_k, u_k) + w_k$$

$$\text{Equação de saída: } z_k = h(x_k) + v_k$$

2. A trajetória nominal é conhecida ao longo do tempo:

$$\bar{x}_{k+1} = f(\bar{x}_k, u_k) \quad (67)$$

$$\bar{z}_k = h(\bar{x}_k) \quad (68)$$

3. A cada passo, computar as seguintes matrizes de derivadas parciais, avaliadas no estado nominal:

$$A_k = f'(\bar{x}_k, u_k) \quad (69)$$

$$H_k = h'(\bar{x}_k) \quad (70)$$

onde as derivadas das equações acima são tomadas em relação a  $x_k$ .

4. Definir  $\Delta z_k$  como a diferença entre a medição atual  $z_k$  e a medição nominal  $\bar{z}_k$ :

$$\begin{aligned} \Delta z_k &= z_k - \bar{z}_k \\ &= z_k - h(\bar{x}_k) \end{aligned} \quad (71)$$

5. Executar as seguintes equações do filtro de Kalman:

$$K_k = P_k H_k^T (H_k P_k H_k^T + R)^{-1} \quad (72)$$

$$\Delta \hat{x}_{k+1} = A_k \Delta \hat{x}_k + K_k (\Delta z_k - H_k \Delta \hat{x}_k) \quad (73)$$

$$P_{k+1} = A_k (I - K_k H_k) P_k A_k^T + Q \quad (74)$$

$$\hat{x}_{k+1} = \bar{x}_{k+1} + \Delta \hat{x}_{k+1} \quad (75)$$

### 5.4.3 O Filtro de Kalman Estendido

O filtro de Kalman linearizado é eficiente para vários casos, mas possui um problema: é necessário saber a trajetória nominal  $\bar{x}$  ao longo do tempo, segundo

Simon (2006). Para alguns sistemas é possível saber esta trajetória, mas para outros não se tem condições de saber a mesma.

A idéia do filtro de Kalman estendido (FKE) é utilizar a estimativa de  $x$  como a trajetória nominal no filtro de Kalman linearizado, ou seja, faz-se  $\bar{x}$  igual a  $\hat{x}$ . Trata-se de uma inteligente aproximação por variabilidade estatística (*bootstrap*) da estimação do estado, partindo da trajetória nominal para estimar  $x$  e então usando este valor como a nova trajetória nominal. Assim, o seguinte algoritmo resume o funcionamento do FKE, após as substituições nas equações do filtro de Kalman linearizado:

1. As equações do sistema são dadas por:

$$\text{Equação de estado: } x_{k+1} = f(x_k, u_k) + w_k$$

$$\text{Equação de saída: } z_k = h(x_k) + v_k$$

2. A cada passo, computar as seguintes matrizes de derivadas parciais, avaliadas no estado nominal:

$$A_k = f'(\hat{x}_k, u_k) \quad (76)$$

$$H_k = h'(\hat{x}_k) \quad (77)$$

onde as derivadas são tomadas em relação a  $x_k$ , e então avaliadas com  $x_k = \hat{x}_k$ .

3. Executar as seguintes equações do filtro de Kalman:

$$K_k = P_k H_k^T (H_k P_k H_k^T + R)^{-1} \quad (78)$$

$$\hat{x}_{k+1} = f(\hat{x}_k, u_k) + K_k [z_k - h(\hat{x}_k)] \quad (79)$$

$$P_{k+1} = A_k (I - K_k H_k) P_k A_k^T + Q \quad (80)$$

## **6 O Filtro de Kalman como apoio ao *Mean Shift***

Baseado em todo o estudo realizado acerca das ferramentas de *mean shift* (MS) e do filtro de Kalman (FK), além de diversos conceitos teóricos de Visão Computacional, este capítulo contém informações práticas sobre os problemas encontrados na fase inicial de desenvolvimento do sistema Kinesis, bem como propõe um algoritmo para união destas ferramentas de forma a melhorar o desempenho do sistema frente a tais diversidades. Este algoritmo será descrito passo a passo e dividido em seções, contendo ações a serem tomadas e explicações sobre as mesmas.

Testes parciais realizados em paralelo a este trabalho mostraram necessidade de um estudo muito mais aprofundado a respeito de modelos de movimentação dos atletas, para uma definição mais apurada das matrizes de transição, covariância e ruído, presentes no filtro de Kalman. Devido a esta complexidade, seria ideal a realização de um segundo projeto com foco central nesta modelagem para se obter resultados práticos satisfatórios a partir do algoritmo que será proposto neste capítulo.

### **6.1 Problemas de Oclusão e Deformação do Alvo**

Conforme citado anteriormente, os principais problemas encontrados nos testes da primeira versão do sistema Kinesis tratam-se dos problemas de oclusão quase total e total do alvo, e deformação acentuada do alvo. A oclusão do alvo é caracterizada no momento em que um atleta é coberto por um objeto ou outra pessoa no ângulo de visão captado pela câmera. O grau de oclusão pode ser parcial, quase total ou total, de acordo com a parcela do corpo do atleta que permanece visível no momento da oclusão.

Em um sistema como o Kinesis, que trabalha a partir da avaliação de semelhança a partir de uma função de densidade de probabilidade, no momento em que ocorre a oclusão, o alvo do rastreamento tem seu perfil e histogramas alterados e, portanto, afeta o processo realizado pelo algoritmo de *mean shift*. Isto ocorre porque no momento da oclusão o MS compara o histograma do *frame* anterior, correspondente ao atleta que vinha sendo rastreado, com o do *frame* atual, pertencente ao atleta que passa a estar na frente do primeiro, pelo ângulo de visão da câmera. Essa diferença minimiza o coeficiente de Bhattacharyya, induzindo ao erro no rastreamento, que provavelmente vai passar a ter como novo alvo o atleta que efetuou a oclusão sobre o antigo alvo.

Em alguns casos de oclusão parcial, o sistema consegue manter o rastreamento correto, ao contrário dos casos de oclusão quase total ou total. A Fig. 17 ilustra o problema de oclusão quase total do alvo, enquanto a Fig. 18 mostra o caso de oclusão total. Estas imagens foram obtidas durante os testes iniciais do Kinesis, realizados por Schnack (2005). Os *frames* nos quais as oclusões ocorrem possuem uma seta indicativa apontando para o alvo do rastreamento.



Figura 17: Oclusão quase total do alvo.

Fonte: SCHNACK, M., 2006.

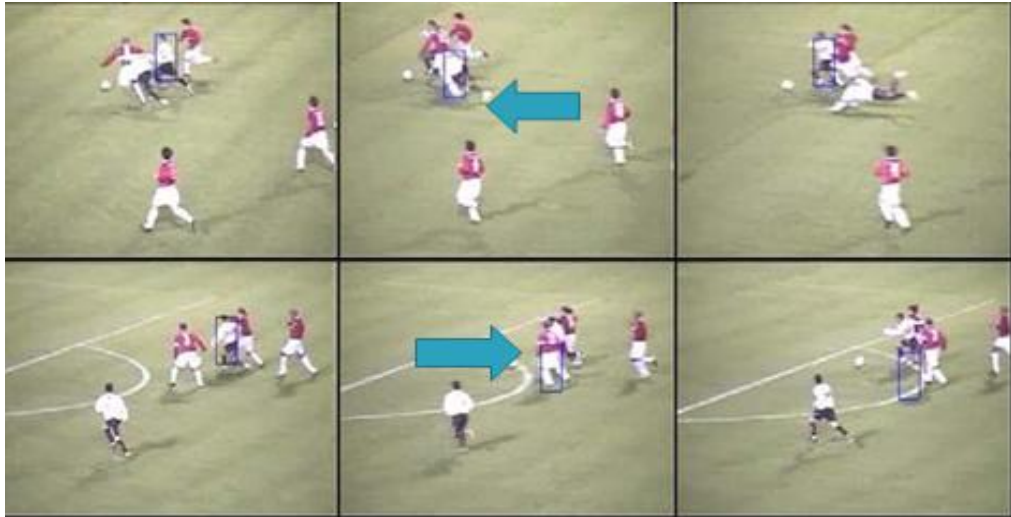


Figura 18: Oclusão total do alvo.

Fonte: SCHNACK, M., 2006.

Quanto ao problema de deformação do alvo, este ocorre quando o atleta sofre alteração em sua forma padrão. Alguns exemplos seriam situações em que o atleta cai, agacha, rola, curva acentuadamente alguma parte do corpo, entre outras. Pode ser leve ou acentuada, dependendo da nova forma assumida pelo atleta. Quando isto acontece, o histograma modelo acaba sendo comparado com um novo histograma que contém informações ruins para o rastreamento, como o fundo da imagem, na maioria dos casos. Em outros casos, o novo formato aparente do jogador pode exceder o limite da região definida como modelo, fazendo com que informações do seu perfil não sejam totalmente incluídas no histograma da região observada como, por exemplo, no caso de um jogador deitado. Assim, o sistema acaba por errar o rastreamento a partir do momento da deformação, na grande maioria dos casos, principalmente quando esta for acentuada.

## 6.2 Motivos para Inclusão do Filtro de Kalman

Pode-se citar algumas vantagens na inclusão do filtro de Kalman no sistema Kinesis, como apoio ao rastreamento realizado pelo algoritmo de *mean shift*. São algumas delas:

- Nos casos de oclusão citados anteriormente, o FK estima a próxima posição do alvo baseado na sua trajetória ao longo do tempo. Sendo

assim, consegue solucionar este problema do *mean shift*, em vários casos;

- O FK estima uma posição ótima para o alvo, para a qual o centro da seleção do alvo é redirecionado, em caso de acerto, ou para a qual o MS aponta como centro da região de interesse. Isto faz com que não seja preciso repetir várias iterações do MS a cada *frame*, ganhando em rapidez, na maioria dos casos;
- O rastreamento direcionado pelas medições do FK consegue ser prorrogado por vários *frames* consecutivos nos quais ocorre oclusão do alvo, devido às informações anteriores da trajetória do atleta. Anteriormente, o MS não conseguia manter sozinho este rastreamento, sendo induzido ao erro;
- Nos casos em que dois atletas com uniforme semelhante ficam próximos um do outro em um determinado *frame*, o MS possui maior probabilidade de errar o rastreamento, devido à semelhança dos histogramas dos dois atletas. Já o FK, a partir do histórico de posições anteriores do modelo do alvo, é capaz de estimar a posição deste mesmo que um atleta com uniforme parecido se aproxime, aumentando a possibilidade de sucesso no rastreamento.

Porém, cabe ressaltar que apesar de todas as vantagens citadas, o filtro de Kalman só obtém sucesso se for muito bem calibrado e modelado no sistema. Ainda assim, uma desvantagem é que ele facilmente erraria a predição, no caso de uma mudança brusca de trajetória realizada por um atleta, principalmente logo após um caso de oclusão. Neste caso, em especial, espera-se que o *mean shift* consiga corrigir o erro, a partir da varredura da região, baseando-se na semelhança do histograma do candidato do alvo em relação ao do modelo.

### 6.3 Algoritmo para Adaptação do Filtro de Kalman ao Kinesis

Com base nas técnicas estudadas, este trabalho propõe um algoritmo para inclusão do filtro de Kalman no sistema Kinesis, como apoio ao *mean shift*. Este algoritmo será descrito em seções contendo as ações a serem tomadas para sua aplicação na prática. A revisão bibliográfica realizada, juntamente com um *feedback*



proveniente de alguns testes realizados na ferramenta MATLAB reforçam a credibilidade do algoritmo, desde que seja aplicado com uma modelagem correta, conforme será explicado mais adiante. A figura 19 ilustra, em um fluxograma, os passos realizados pelo algoritmo proposto.

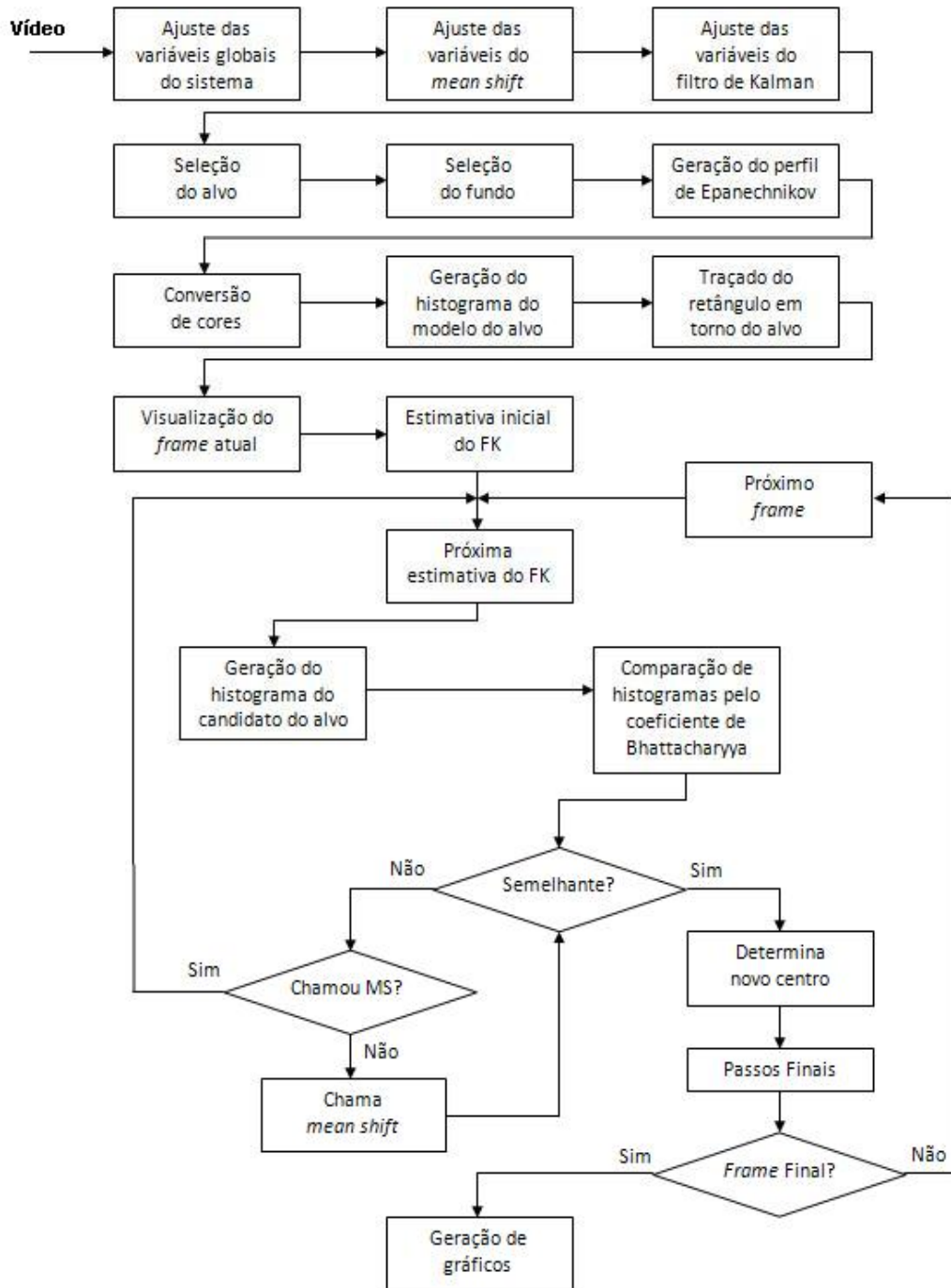


Figura 19: Fluxograma do algoritmo de união do FK e MS.

Basicamente, o sistema Kinesis trabalha a partir de um vídeo, do qual o primeiro *frame* é visualizado na tela, para que o usuário selecione o alvo do rastreamento e uma parte do fundo da cena. Para isto, algumas variáveis referentes ao processo devem ser ajustadas previamente. Na seqüência, o algoritmo reproduz os *frames* subseqüentes do vídeo, realizando o rastreamento do alvo selecionado, marcando-o com um retângulo. Este rastreamento é direcionado pelo filtro de Kalman, com apoio do *mean shift* quando o candidato do alvo não é suficientemente semelhante ao modelo do alvo, de acordo com a comparação pelo coeficiente de Bhattacharyya. Os passos realizados pelo algoritmo são descritos detalhadamente a seguir.

### 6.3.1 Ajuste Inicial das Variáveis Globais do Sistema

Inicialmente, devem ser escolhidos os valores iniciais para as seguintes variáveis globais do sistema:

- Arquivo de vídeo: trata-se da escolha do vídeo utilizado para o rastreamento, o qual é carregado a partir de um arquivo armazenado em disco.
- *Frame* inicial: deve-se escolher qual *frame* do vídeo será utilizado como inicial, para seleção do alvo a ser rastreado. Geralmente esta variável é ajustada para o primeiro *frame* do vídeo.
- Número de *frames*: corresponde ao ajuste de quantos *frames* do vídeo serão reproduzidos ao longo do rastreamento.

### 6.3.2 Ajuste Inicial das Variáveis do *Mean Shift*

- Número de *bins*: trata-se da escolha de quantos *bins* serão utilizados para a construção do histograma do alvo e do fundo.

### 6.3.3 Ajuste Inicial das Variáveis do Filtro de Kalman

A descrição dos ajustes das variáveis iniciais para o FK foi baseado em Brookner (1998) e Lin (2006).

- Passo de tempo: deve-se ajustar o valor correspondente ao espaço de tempo utilizado pelo algoritmo.
- Vetor de estado ( $x$ ): cada elemento deste vetor deve ser ajustado para armazenar uma variável do sistema. No caso do Kinesis, devem ser armazenadas a posição e a velocidade do alvo para ambos os eixos ( $x, y$ ). Preferencialmente, um *array* de vetores de estado deve ser criado, para armazenar os resultados das medições a cada iteração. Os valores para a posição inicial desse *array* de vetores correspondem ao valor do centro em  $x$  e em  $y$  do alvo selecionado, e à velocidade inicial do alvo, na direção de cada eixo.
- Matriz de transição ( $A$ ): o ajuste da matriz de transição deve ser baseado no modelo de movimentação do alvo a ser rastreado. No caso do Kinesis, onde o alvo é um atleta atuando em um esporte, o modelo deve levar em conta a posição e a velocidade deste atleta.
- Vetor de controle ( $Bu$ ): determina o movimento padrão do alvo a ser rastreado, induzindo a estimação do filtro em uma determinada direção. É ajustado de acordo com a aplicação.
- Ruído de medição ( $w$ ): corresponde ao ruído existente na imagem com a qual o algoritmo está trabalhando.
- Matriz de observação ( $H$ ): componente da equação de observação do sistema, normalmente ajustada como uma matriz identidade  $2 \times 2$ . No caso de trabalhar-se com vetores de estado com mais do que duas variáveis, normalmente as outras colunas desta matriz são preenchidas com o valor 0.
- Ruído de observação ( $v$ ): corresponde ao ruído proveniente dos aparelhos utilizados para capturação das imagens do vídeo. Representa o erro na observação a cada iteração.
- Ruído de covariância do processo ( $Q$ ): representa a magnitude da incerteza da trajetória do alvo e deve ser ajustado com base no ruído de medição.
- Ruído de covariância da observação ( $R$ ): é relacionado com a eficácia das medições das ferramentas utilizada no sistema. É ajustado com

base no ruído de observação. Trata-se do ajuste mais crítico do filtro de Kalman, e indispensável para um funcionamento correto.

#### 6.3.4 Seleção do Alvo

Como dito anteriormente, o algoritmo gera a imagem inicial de um vídeo escolhido e aguarda a interação do usuário, que deve selecionar o alvo a ser rastreado. O algoritmo armazena as informações referentes ao tamanho do retângulo de seleção e a posição  $(x,y)$  do centro deste retângulo. Esta seleção corresponde ao modelo do alvo do rastreamento. A Fig. 20 mostra um exemplo de seleção do alvo no sistema Kinesis, a partir do algoritmo inicial descrito por Schnack (2005), usando a ferramenta MATLAB para simulação.

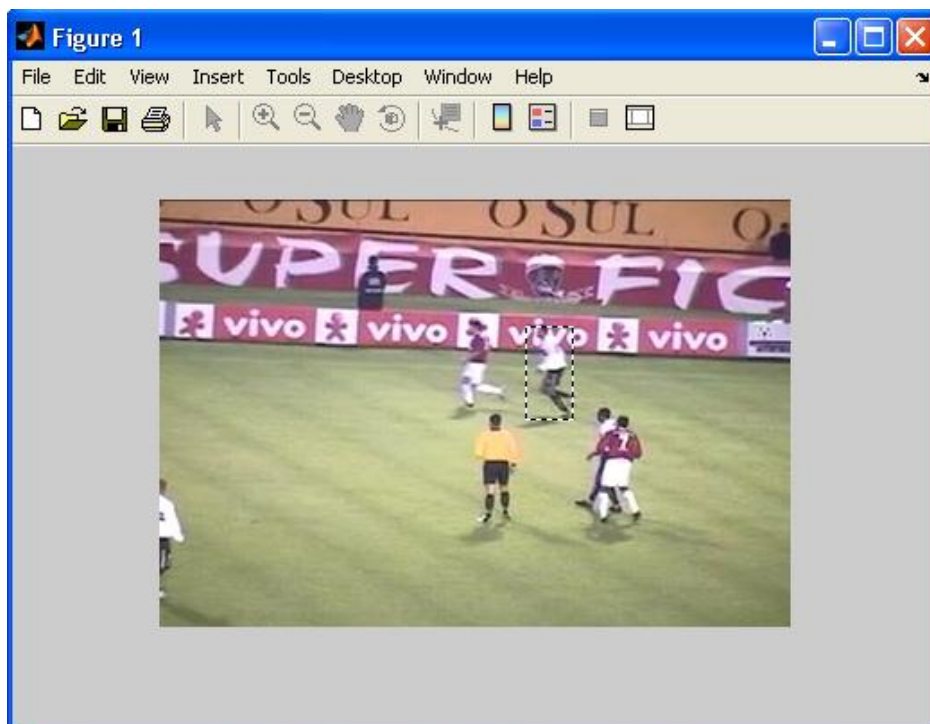


Figura 20: Simulação do Kinesis na ferramenta MATLAB – Tela de seleção do alvo.

#### 6.3.5 Seleção do Fundo

Semelhante ao item anterior, o usuário deve selecionar uma amostra do fundo da imagem. O algoritmo armazena informações referentes ao centro e

tamanho do fundo. A Fig. 21 mostra um exemplo de seleção do fundo no sistema Kinesis.

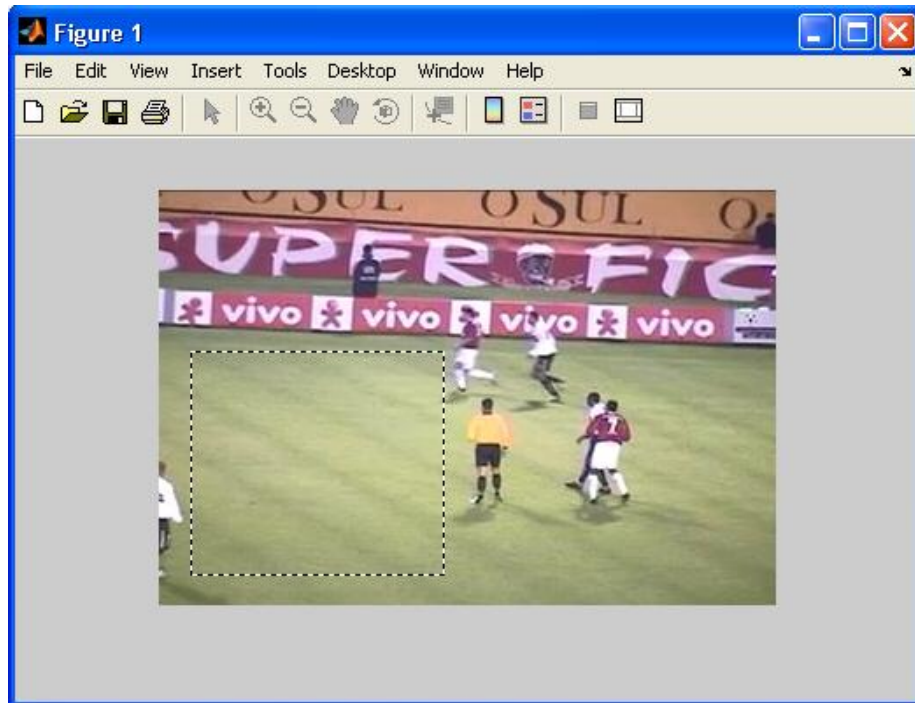


Figura 21: Simulação do Kinesis na ferramenta MATLAB – Tela de seleção do fundo.

### 6.3.6 Geração do Perfil Epanechnikov do Modelo do Alvo

O perfil Epanechnikov, definido na seção 4.2.1, deve ser gerado para minimizar a medida do erro quadrado entre a densidade de cores da região selecionada como modelo do alvo e a estimativa da mesma. Sendo assim, é informação indispensável para a construção dos histogramas.

### 6.3.7 Conversão de Cores

Pelos motivos explicados na seção 4.5.2, o sistema de cores HSB é utilizado no Kinesis. Como a imagem captada pelo vídeo é carregada no formato padrão RGB, uma conversão entre os sistemas é realizada, de acordo com as equações descritas na seção 4.5.3.

### 6.3.8 Geração do Histograma do Modelo do Alvo

Nesta etapa é gerado o histograma do modelo do alvo, para comparação de semelhança com os candidatos do alvo, a partir do coeficiente de Bhattacharyya. Esta comparação é descrita mais adiante no algoritmo. Para a geração do histograma, são utilizadas as seguintes informações: centro, tamanho e perfil Epanechnikov do modelo do alvo, imagem resultante da conversão de cores, e número de *bins*. A Fig. 22 mostra um exemplo de histograma gerado para o modelo do alvo na simulação do sistema Kinesis.

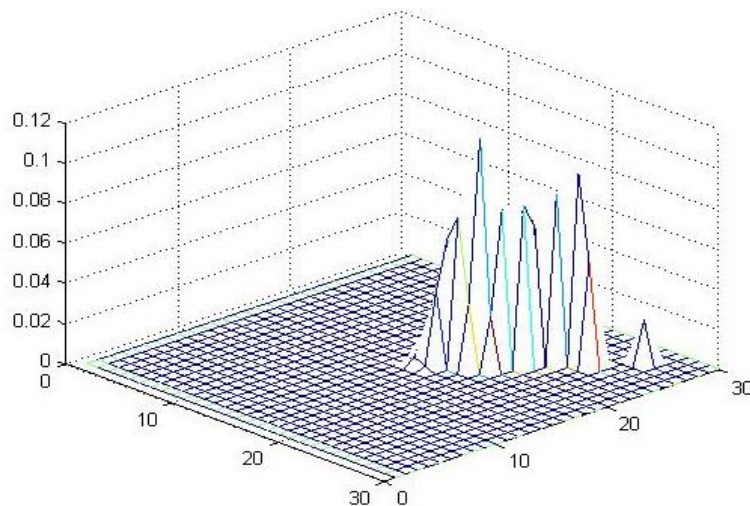


Figura 22: Histograma do modelo do alvo gerado pelo sistema Kinesis.

### 6.3.9 Traçado do Retângulo em Torno do Alvo

O algoritmo traça um retângulo em torno do alvo do rastreamento, com tamanho idêntico ao retângulo de seleção traçado pelo usuário. São utilizadas informações da posição atual do *array* de vetores de estado, de forma que os centros deste vetor são utilizados para traçar o retângulo.

### 6.3.10 Visualização do *Frame* Atual

Após todos os passos anteriores, a imagem referente ao *frame* atual do vídeo no instante de tempo corrente é amostrada na tela. O retângulo traçado no

passo anterior aparece em torno do candidato alvo do rastreamento, a partir do qual se pode avaliar visualmente se o rastreamento está ocorrendo de forma satisfatória.

### **6.3.11 Estimativa Inicial do Filtro de Kalman**

Conforme estudado anteriormente, a partir de um estado inicial de  $x$ , o FK realiza a primeira estimativa, utilizando o conjunto de equações de predição e correção, apresentados na seção 5.3.3. A partir de um estado inicial, e de uma primeira estimativa, é possível estimar os próximos estados do sistema.

### **6.3.12 Laço de Iteração**

Realizados os passos anteriores, o algoritmo entra em um laço de iteração, controlado pelo *frame* atual do vídeo. Neste laço, inicialmente, são repetidas as tarefas de amostragem do *frame* atual na tela, conversão do sistema de cores e, então, novamente a estimação do FK é realizada, para o novo estado.

### **6.3.13 Próxima Estimativa do Filtro de Kalman**

A nova estimativa do FK é tomada com base na estimativa inicial e utilizando novamente o conjunto de equações citado anteriormente. Novos valores são armazenados na posição seguinte do *array* de vetores de estado, então o ganho de Kalman é computado e a covariância e o erro de medição são atualizados.

### **6.3.14 Geração do Histograma do Candidato do Alvo**

A estimação do FK fornece um novo valor de centro em  $x$  e  $y$  para o candidato do alvo. Resta agora saber se este candidato é semelhante ao modelo do alvo, para manter a nova posição, ou se é necessário chamar o procedimento de *mean shift* para varrer a região em busca de um melhor ponto para este centro. Portanto, o histograma do candidato do alvo precisa ser gerado para comparação com o histograma do modelo. Para isto, são utilizadas informações sobre os valores

dados pela medição do FK para o novo centro, o tamanho do modelo do candidato, a imagem após conversão de cores, o perfil Epanechnikov do modelo do alvo e o número de *bins* escolhido inicialmente.

### 6.3.15 Comparação de Histogramas pelo Coeficiente de Bhattacharyya

Com ambos os histogramas (do modelo do alvo e do candidato) gerados, o algoritmo então utiliza o coeficiente de Bhattacharyya para medir a distância entre as duas distribuições de probabilidade. Conforme os cálculos apresentados na seção 4.3 para esta métrica, chega-se a um valor numérico que representa a semelhança entre os dois histogramas. Quanto maior for a distância entre as duas distribuições, mais semelhantes elas são. Um valor baixo de distância significa que o ponto predito pelo FK não é uma boa escolha. Isto é decidido através de uma faixa determinada pelo usuário como limites para uma boa ou má escolha, com base neste valor de distância.

### 6.3.16 Chamando o Procedimento de *Mean Shift*

No caso da predição do FK não se tratar de uma boa escolha, o procedimento de *mean shift* é chamado. Assume-se o ponto de centro dado pelo FK como centro da região de interesse para o MS, e então o processo de busca à região com maior densidade de probabilidade relacionada ao histograma do modelo do alvo selecionado pelo usuário. O número de iterações por *frame* e o critério de parada para o MS deve ser ajustado pelo usuário. No caso de o MS não encontrar a melhor posição para o novo centro dentro desta região de interesse, deve-se retornar ao passo anterior do laço, repetindo a estimação do FK, em busca de um centro diferente. Caso a convergência seja alcançada, então o novo centro é dado pelo MS.

### 6.3.17 Determinando o Novo Centro

De acordo com os passos anteriores, o novo centro depende diretamente da semelhança apontada pelo coeficiente de Bhattacharyya. Se a distância estiver



acima do limite mínimo para aceitação, então o novo centro é dado pelo FK. Caso contrário, o novo centro é fornecido pelo MS.

### 6.3.18 Passos Finais

Com o novo centro estabelecido, o retângulo de seleção do alvo do rastreamento é traçado sobre a imagem do *frame* atual e este, então, é mostrado na tela. Todos os passos posteriores à seção 6.3.12 (laço de iteração) são repetidos até que se atinja o número de *frames* limite estabelecido pelo usuário.

### 6.3.19 Geração de Gráficos

O usuário pode gerar gráficos dos resultados obtidos pelo algoritmo acima descrito. Estes gráficos, para este tipo de aplicação, seriam as posições real, estimada e medida (provenientes do FK em conjunto com o MS), além de outras informações, como velocidade, aceleração e o coeficiente de Bhattacharyya ao longo do tempo. Estas informações são levadas em conta na geração dos dados estatísticos a respeito dos atletas, tratando-se de um dos objetivos do sistema Kinesis. Um exemplo de geração de gráfico é mostrado na Fig. 23, que representa a velocidade em *pixels* por *frame* de um alvo rastreado nos testes iniciais do Kinesis.

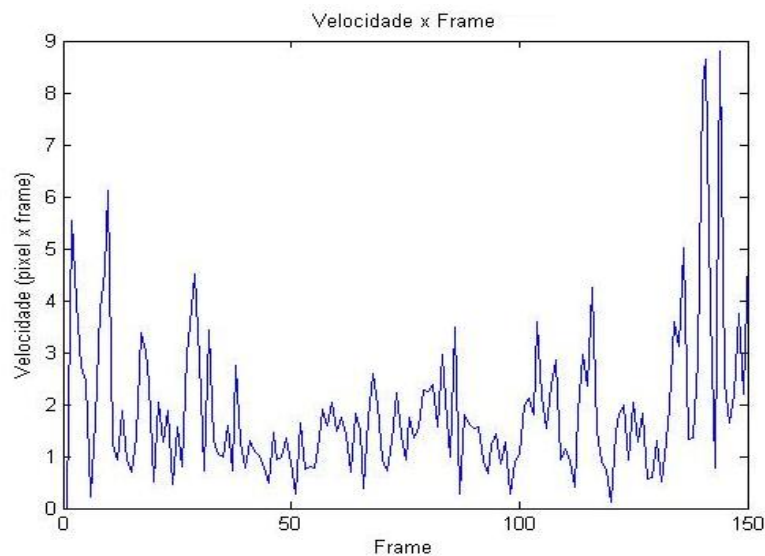


Figura 23: Gráfico da velocidade do alvo rastreado ao longo dos *frames*.

## 7 Conclusão

O estudo realizado ao longo deste trabalho mostra que a técnica do filtro de Kalman (FK), incluindo suas extensões, é bastante ampla e trata-se, atualmente, de uma das principais ferramentas aplicadas a problemas de navegação e controle. Sendo assim, sua inclusão no sistema Kinesis em conjunto com o *mean shift* (MS) é de grande importância, frente às dificuldades encontradas nos testes iniciais.

Porém, o MS precisou ser estudado a fundo para que se pudesse chegar à proposta de um algoritmo mais eficiente, passível de implementação futura. Além disto, conceitos introdutórios de visão computacional e rastreamento foram estudados e inclusos na abordagem deste trabalho.

Portanto, este trabalho parte do conceito de visão computacional, indispensável para o entendimento do paralelo entre a visão humana e a visão de máquina. Outro conceito fundamental é o de aprendizado de máquina, base para qualquer ferramenta que se baseie em conhecimento anterior para estimar um resultado posteriormente. Tanto o filtro de Kalman quanto o *mean shift* são exemplos destas ferramentas.

Tais técnicas pertencem à classe de rastreamento baseado em transformação e objetiva-se aplicá-las ao sistema Kinesis de forma a constituir uma ferramenta de rastreamento baseado em imagens. Com base na revisão bibliográfica realizada, conclui-se que a aplicação destas técnicas requer custo computacional baixo, em relação às outras técnicas discutidas. Isto se deve ao fato de apenas se fazer necessária a utilização de uma câmera e um dispositivo de processamento de imagens, para a realização do rastreamento, a partir de duas técnicas de rápido funcionamento.

O sistema Kinesis, em sua versão inicial foi estudado profundamente, possibilitando o entendimento em detalhes do algoritmo de *mean shift*. Conceitos de

processamento de imagens como geração de histogramas, teorias Gaussianas, sistemas de cores, definição de funções de densidade de probabilidade (PDF) e comparação de PDFs através do coeficiente de Bhattacharyya fizeram parte do processo.

Pode-se afirmar que o MS é um poderoso algoritmo para avaliação de movimento de um objeto a partir de imagens, para o qual não se faz necessária a avaliação por completo da imagem, o que caracteriza sua rapidez de processamento. Porém, se mostrou uma técnica de difícil aplicação isolada em vídeos de jogos de esportes coletivos, como é o caso do sistema Kinesis. Isto porque os casos de oclusão e deformação do alvo são muito constantes em jogos coletivos, o que torna necessária a utilização de uma segunda ferramenta como apoio para estes casos.

Testes parciais paralelos ao desenvolvimento deste projeto, utilizando a ferramenta MATLAB, mostraram que a aplicação do MS necessita de uma avaliação do número de *bins* utilizados na construção dos histogramas, adaptada a cada caso. Se o número de *bins* for muito grande, a diferença entre o histograma do modelo e do candidato do alvo fornecida pelo coeficiente de Bhattacharyya pode sofrer variações devido a uma alta granularidade da amostragem, levando a uma discrepância maior entre estes dois histogramas, influenciada pela variação de cores e luminosidade. Como resultado destes testes parciais, chegou-se à conclusão de que 30 *bins* são suficientes para se definir uma boa faixa para a normalização dos histogramas.

Após avaliação de suas vantagens, o FK foi proposto, então, como novo alicerce do rastreamento do Kinesis, por ser mais abrangente que o *mean shift*, podendo gerar resultados satisfatórios por si só, até mesmo para os casos críticos. Assim, o MS passa a ser uma alternativa ao FK, quando este não fornece uma estimativa que obtenha uma semelhança a nível satisfatório entre o candidato do alvo e o modelo do alvo no rastreamento.

Portanto, o filtro de Kalman padrão foi estudado a fundo, bem como algumas de suas extensões para problemas não-lineares, como é o caso do Kinesis. Porém, o FK necessita de uma modelagem extremamente detalhista de cada matriz, vetor ou variável simples, presente em suas equações de predição e correção. Para isto, conceitos de Álgebra Linear foram revistos, em especial operações com matrizes.

Dentre todos os modelos a serem construídos, conclui-se que o ruído de covariância da observação ( $R$ ) é o fator mais crítico para a aplicação prática do FK. Para se obter uma modelagem ótima deste ruído, se faz necessário um conhecimento profundo sobre a dinâmica de movimentação dos atletas, em diversas situações, além de uma integração com todos os outros modelos, que também precisam ser definidos corretamente. Testes parciais na ferramenta MATLAB comprovam a conclusão de que, idealmente, esta modelagem avançada das matrizes do FK, em especial do ruído de covariância da observação, necessitaria ser desenvolvida em um projeto à parte, por motivos de complexidade, para se obter, na prática, o rastreamento correto a partir do FK. Estes testes mostraram também que os cálculos do FK estão sendo realizados corretamente, bem como todo o procedimento de *mean shift* e análise do coeficiente de Bhattacharyya. Porém, esta complexidade de modelagem do ruído de covariância de observação dificultou o funcionamento correto do rastreamento nos testes parciais realizados paralelamente a este trabalho.

Dada a não-linearidade da aplicação do Kinesis, conclui-se que o ideal seria a aplicação do filtro de Kalman estendido. Esta extensão é mais complexa de se implementar do que o filtro padrão, porém fornece resultados muito melhores para o problema em questão.

Baseando-se nisto, este trabalho apresentou uma proposta de algoritmo passo a passo centrada em todo o estudo realizado, e fortalecida pelos testes parciais realizados paralelamente a este projeto. Espera-se, portanto, que a união das técnicas do filtro de Kalman e do *mean shift* seja efetiva, já que uma técnica é capaz de suprir as necessidades da outra, nas diversas situações a que o sistema possa ser submetido.

Quanto a algumas destas situações em específico, conclui-se que o algoritmo aparentemente apresenta um bom nível de recuperação de erros de rastreamento, exceto na situação de oclusão entre jogadores com uniformes parecidos, devido à utilização de informação de cores para verossimilhança, o que levaria à comparação de dois histogramas muito parecidos, como comentado anteriormente. Isto poderia induzir ao erro no rastreamento.

Já nas demais situações de oclusão, o FK se mostrou uma ferramenta bastante robusta, devido ao fato de que consegue manter seu rastreamento ao longo de vários *frames* em que o alvo se encontre ocluso, a partir da estimação

baseada no histórico da trajetória do atleta. Eventualmente erros podem acontecer, caso este atleta mude bruscamente sua trajetória enquanto estiver ocluso, o que pode ser corrigido pelo auxílio do *mean shift*.

Em um escopo geral, conclui-se que o estudo aprofundado de diversos conceitos e técnicas é de grande valia como complemento à grade curricular do curso de Bacharelado em Ciência da Computação da Universidade Federal de Pelotas (UFPEL), que não inclui a maioria dos conceitos estudados neste trabalho. Este trabalho possibilitou um bom nível de aprendizado na área de visão computacional, com foco nos problemas de rastreamento. Trata-se de técnicas interessantes e de grande importância prática no panorama atual da computação aplicada à área esportiva.

Além disso, mais um passo foi dado em direção a se desenvolver uma ferramenta útil aos profissionais da área esportiva, de forma a potencializar o desempenho dos atletas em ação. A seguir, trabalhos futuros são indicados, visando o objetivo final de um sistema capaz de avaliar atletas em tempo real, fornecendo dados estatísticos a seus treinadores e preparadores físicos.

## 7.1 Trabalhos Futuros

De forma a dar seguimento ao desenvolvimento do sistema Kinesis, alguns trabalhos futuros podem ser indicados. São eles:

- Modelagem aprofundada das matrizes componentes do filtro de Kalman, em especial do ruído de covariância de observação, por motivos explicados anteriormente;
- Realização de uma série extensa de testes com o sistema Kinesis, avaliando o desempenho deste nas mais diversas situações presentes nos vídeos de jogos de esportes coletivos;
- Histograma adaptativo: um problema existente no rastreamento em uma partida de futebol é a constante mudança no visual dos jogadores. Assim sendo, seria válido modificar o modelo do alvo para refletir essas possíveis alterações. Dessa forma, ao finalizar o algoritmo de *mean shift* e ter a localização que maximiza o coeficiente de Bhattacharyya no frame corrente, o histograma do candidato do alvo teria suas

informações de alguma forma inseridas no histograma do alvo. O coeficiente de Bhattacharyya deve funcionar como peso nessa inserção, visto que não é desejável que dados que não pertençam ao alvo sejam adicionados ao seu modelo. Essa preocupação visa evitar que informações de outros objetos, no caso de possíveis oclusões do alvo, sejam consideradas como parte do objeto a ser rastreado

- Abordagem Multi-Modelo: atualmente, o sistema Kinesis efetua o rastreamento em apenas um alvo a cada execução. Uma idéia para tornar o sistema mais atraente e prático seria a expansão dele para suportar o rastreamento de mais de um alvo simultaneamente. Assim sendo, em uma partida de esporte coletivo poderia se efetuar o rastreamento em todos os jogadores da equipe em treinamento e até nos jogadores adversários, visando a melhor análise da performance da equipe e dos oponentes. Como o sistema recria a movimentação do alvo, com o rastreamento de todos os participantes de uma partida seria possível reconstituí-la totalmente. Além de aumentar a capacidade de avaliação da partida, essa recriação mais simplória poderia ser transmitida para outros dispositivos mais facilmente. Isso possibilitaria uma transmissão ao vivo do evento via aparelhos de telefonia móvel, por exemplo.
- Implementação do filtro de partículas ou técnicas de Monte Carlo: adequados a sistemas não-lineares e de rastreamento de vários alvos.
- Tempo Real: objetivo máximo do sistema Kinesis, necessita de diversas adaptações nos algoritmos para ser implementado, partindo da migração da linguagem da ferramenta MATLAB para outra linguagem de programação mais rápida, bem como um refinamento nos procedimentos de *mean shift* e filtro de Kalman (ou outro filtro) com relação ao número de iterações realizadas até que se atinja o resultado esperado. A grande maioria dos trabalhos futuros propostos por este trabalho seria de grande auxílio para a instauração do tempo real.

## Referências

ARCHAMBEAU, C.; VERLEYSEN, M. Fully Nonparametric Probability Density Function Estimation with Finite Gaussian Mixture Models. **ICAPR'2003 proceedings - 5th International Conference on Advances in Pattern Recognition**. India, 2003. p.81-84.

BAR-SHALOM, Y. **Estimation with Applications to Tracking and Navigation**. John Wiley & Sons, 2001. 584p.

BIGUN, J. **Vision With Direction: A Systematic Introduction to Image Processing and Computer Vision**. 1.ed. Springer, 2006. 396p.

BROOKNER, E. **Tracking and Kalman Filtering Made Easy**. John Wiley & Sons, 1998. 492p.

BROWN, R; HWANG, P. **Introduction to Random Signals and Applied Kalman Filtering**. 3.ed. John Wiley & Sons, 1997. 496p.

CASTELLANOS, J. A. G. **Estimação de Velocidade do Motor com Controle Vetorial sem Sensor, Utilizando Filtro Estendido de Kalman com Estimação da Covariância dos Ruídos**. 2004. Tese de Doutorado (Pós-Graduação em Engenharia Elétrica)- Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.

COMANICIU, D.; MEER, P. Mean Shift Analysis and Applications. **IEEE Int. Conf. Computer Vision (ICCV'99)**. Kerkyra, Greece, 1197-1203, 1999.

COMANICIU, D.; MEER, P. Mean Shift: A Robust Approach Toward Feature Space Analysis. **IEEE transactions on pattern analysis and machine intelligence**, vol. 24, No. 5, 2002.

COMANICIU, D.; MEER, P. Real-Time Tracking of Non-Rigid Objects Using Mean Shift. **IEEE Conf. Computer Vision and Pattern Recognition (CVPR'00)**. Hilton Head Island, South Carolina, vol. 2, 142-149, 2000.

COMANICIU, D.; RAMESH, V.; MEER, P. *Kernel-Based Object Tracking*, **IEEE Trans. Pattern Analysis Machine Intell.**, vol. 25, No. 5, 564-575, 2003.

FORSYTH, D. A.; PONCE, J. **Computer Vision: A Modern Approach**. 1 ed. Prentice Hall, 2002. 809p.

FUKUNAGA, K.; HOSTETLER, L. D. The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition, **IEEE Trans. Info. Theory**, vol. IT-21, 32-40, 1975.

GABRIEL, P. F.; VERLY, J. G.; PIATER, J. H.; GENON, A. The State of the Art in Multiple Object Tracking Under Occlusion in Video Sequences. **Advanced Concepts for Intelligent Vision Systems**, p. 166–173, 2003.

GREWAL, M; ANDREWS, A. **Kalman Filtering: Theory and Practice Using MATLAB**. 2.ed. John Wiley & Sons, 2001.

HARITAOGLU, I. **A Real Time System for Detection and Tracking of People and Recognizing Their Activities**. 1998. 35f. *PhD Proposal (PhD in Computer Science)*-Computer Science Dept.,University of Maryland, College Park.

JÄHNE, B. **Digital Image Processing**. 5.ed. Springer, 2002. 585p.

JÄHNE, B.; HAUßECKER, H. **Computer Vision and Applications – A Guide for Students and Practitioners**. 1. ed. Academic Press, 2000. 679p.

KALMAN, R. E. A New Approach to linear filtering and prediction problems. **Transactions of the ASME - Journal of Basic Engineering**, 35-45, 1960.

LEANDRO, J. Filtro de Kalman e Rastreamento em Seqüências de Vídeo. 2006.  
Disponível em:  
<[http://www.vision.ime.usp.br/creativision/events/vision/files/2006/26\\_07.pdf](http://www.vision.ime.usp.br/creativision/events/vision/files/2006/26_07.pdf)>  
Acesso em: 24 jun. 2007.

LIN, I-Chen. Human Computer Interaction – Vision-Based Controls (A). 2006.  
Disponível em: <[http:// caig.cs.nctu.edu.tw/course/HCI06/HCI\\_0\\_Preface.pdf](http://caig.cs.nctu.edu.tw/course/HCI06/HCI_0_Preface.pdf)>.  
Acesso em: 06 ago. 2007.

MISUTA, Milton Shoiti. **Rastreamento Automático de Trajetórias de Jogadores de Futebol por Videogrametria: Validação do Método e Análise dos Resultados**. 2004. Dissertação de Mestrado (Pós-Graduação da Faculdade de Educação Física)-Universidade Estadual de Campinas, Campinas.

NEDOVIĆ, V. Tracking moving video objects using mean-shift algorithm. **IEEE Vehicular Technology Conference 2004-Fall**. Project Report, 2004.

PORIKLI, F.M.; TUZEL, O. Human Body Tracking by Adaptive Background Models and Mean-Shift Analysis. **IEEE International Workshop on Performance Evaluation of Tracking and Surveillance**, 2003.

SCHNACK, Martim Guilherme. **Rastreamento de jogadores em esportes coletivos e extração de métricas para avaliação posterior**. 2005. Projeto de Diplomação (Bacharelado em Ciência da Computação)-Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

SCHUTTER, J.; GEETER, J.; LEFEBVRE, T.; BRUYNINCKX, H. Kalman Filters: A Tutorial. **Journal A**, Vol. 40 (4), 1999, 52-59.

SEBE, N.; COHEN, I.; GARG, A.; HUANG, T. S. **Machine Learning in Computer Vision**. 1.ed. Springer, 2005. 240p.

SHAPIRO, L.; STOCKMAN, G. **Computer Vision**. 1.ed. Prentice Hall, 2001. 609p.



FERRUGEM, A., SILVA JÚNIOR, E. P., IDIART, M. A. P., BARONE, D. A. C. A Perceptual User Interface Using Mean Shift. **Lecture Notes in Computer Science**, 2004.

SIMON, D. **Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches**. John Wiley & Sons, 2006.

SIMON, D. Using Nonlinear Kalman Filtering to Estimation Signals. **Embedded Systems Design**, vol. 19, no. 7, pp. 38-53, 2006.

UKRAINITZ, Y; SAREL, B. Mean Shift: Theory and Applications. 2004. Disponível em:  
<[http://www.wisdom.weizmann.ac.il/~deniss/vision\\_spring04/files/mean\\_shift.ppt](http://www.wisdom.weizmann.ac.il/~deniss/vision_spring04/files/mean_shift.ppt)>. Acesso em: 20 mai. 2007.

VELHOTE, M. Filtro de Kalman: A estimação estocástica aplicada em análise de sinais e de imagens. 2005. Disponível em:  
<[ineb.fe.up.pt/bmivic/upcoming/seminars\\_files/kalman.pdf](http://ineb.fe.up.pt/bmivic/upcoming/seminars_files/kalman.pdf)> Acesso em: 13 jun. 2007.

WELCH, G.; BISHOP, G. An Introduction to the Kalman Filter. **Technical Report: TR95-041**, 1999.

WIKIPEDIA, *The Free Encyclopedia*. Kernel (statistics). Disponível em:  
<[en.wikipedia.org/wiki/Kernel\\_\(statistics\)](http://en.wikipedia.org/wiki/Kernel_(statistics))> Acesso em: 03 ago. 2007.