

**GUILHERME MACHADO GAGLIARDI**

**SISTEMA ROBUSTO DE  
ACOMPANHAMENTO DE  
TRAJETÓRIA DE ALVOS MÓVEIS**

Trabalho de Conclusão de Curso apresentado à Escola de  
Engenharia de São Carlos, da Universidade de São Paulo

Curso de Engenharia da Computação

**ORIENTADOR:** Prof. Dr. Marco Henrique Terra

São Carlos  
2014



**GUILHERME MACHADO GAGLIARDI**

**SISTEMA ROBUSTO DE  
ACOMPANHAMENTO DE  
TRAJETÓRIA DE ALVOS MÓVEIS**

Trabalho de Conclusão de Curso apresentado à Escola de  
Engenharia de São Carlos, da Universidade de São Paulo

Curso de Engenharia da Computação

**ORIENTADOR:** Prof. Dr. Marco Henrique Terra

São Carlos  
2014



# FOLHA DE APROVAÇÃO

**Nome:** Guilherme Machado Gagliardi

**Título:** "Sistema robusto de acompanhamento de trajetória de alvos móveis"

**Trabalho de Conclusão de Curso defendido em** 24 / 11 / 2014.

**Comissão Julgadora:**

**Resultado:**

Prof. Associado Marco Henrique Terra  
(Orientador) - SEL/EESC/USP

Aprovado

Prof. Dr. Valdir Grassi Júnior  
SEL/EESC/USP

Aprovado

Prof. Assistente Carlos Goldenberg  
SEL/EESC/USP

APROVADO

**Coordenador do Curso Interunidades - Engenharia de Computação:**

Prof. Associado Evandro Luís Linhari Rodrigues



Aos meus pais, pelo apoio e pela  
paciência.

---





Se queres conhecer o passado,  
examina o presente que é o  
resultado; se queres conhecer o  
futuro, examina o presente que é a  
causa.

---

Confúcio



## Resumo

Este trabalho de conclusão de curso apresenta um sistema de acompanhamento de um alvo móvel em uma sequência de vídeo capturada em tempo real. Para a composição do sistema, são associados o algoritmo de rastreamento por cor *camshift*, uma versão adaptativa do conhecido *mean shift*, e dois algoritmos de predição de estado da linha do filtro de Kalman, sendo um deles uma implementação robusta particularmente adequada a aplicações de tempo real. São fornecidos aspectos introdutórios referentes aos componentes do sistema, iniciando-se com a apresentação do espaço de cores HSV, que, por separar as componentes de cromaticidade e de luminância das cores, é mais adequado do que o tradicional RGB a aplicações como segmentação e rastreamento. A seguir, é descrito o algoritmo de projeção reversa de histograma, utilizado para construir uma distribuição de probabilidade de cor, que indica, para cada pixel de um quadro do vídeo, a probabilidade de que o mesmo pertença ao alvo, conforme representado por um histograma-modelo da componente de matiz de seus pixels. O algoritmo *mean shift* é apresentado como uma maneira de se determinar a região de máxima probabilidade de pertencimento ao alvo na cena, e, na sequência, é mostrada a forma como o *camshift* o aprimora com o ajuste adaptativo do tamanho da janela de busca, tornando o rastreamento mais robusto. Após a descrição dos algoritmos preditores nominal e robusto, segue-se uma visão geral do sistema de rastreamento, com sua interface e sua sequência de atividades. Por fim, são mostrados os resultados de uma avaliação de desempenho conduzida a fim de testar a eficiência computacional do sistema e de comparar os preditores empregados.

**Palavras-chave:** acompanhamento de alvos, filtragem robusta, processamento de imagens.



## Abstract

This text presents a tracking system for a mobile target in a video sequence that is captured in real-time. To compose the system, an association is made between the color-tracking algorithm *camshift*, an adaptive version of the well-known *mean shift*, and two state prediction algorithms similar to the Kalman filter; one of them is a robust implementation, which is suitable for real-time applications. Introductory aspects of the components of the system are presented, beginning with the description of the HSV color space, which, by separating the chrominance and luminance components of the colors, becomes more suitable for applications like segmentation and tracking than the traditional RGB space. Next, the *histogram backprojection* algorithm is described as a way of creating a color probability distribution, which indicates, for each pixel of a frame of the video, the probability that it belongs to the target, which is represented by a model histogram of the hue component of its pixels. The *mean shift* algorithm is presented as a procedure to determine the region with the maximum probability of pertaining to the target in the scene; then it is shown how *camshift* improves it with the adaptive adjustment of the size of the searching window, what makes the tracking more robust. After the description of the nominal and robust prediction algorithms, follows an overview of the tracking system, including its interface and its sequence of activities. Finally, the results of an evaluation of performance are presented, with the objectives of assessing the computational efficiency of the system and of comparing the two estimators.

**Keywords:** target tracking, robust filtering, image processing.



## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Organização do texto . . . . .	2
<b>2</b>	<b>Representação e localização do alvo</b>	<b>3</b>
2.1	O espaço de cores HSV . . . . .	3
2.2	Distribuição de probabilidade de cor . . . . .	7
2.3	O algoritmo <i>mean shift</i> . . . . .	8
2.4	O algoritmo <i>camshift</i> . . . . .	12
<b>3</b>	<b>Filtragem e associação de dados</b>	<b>17</b>
3.1	Estimador nominal . . . . .	20
3.2	Estimador robusto . . . . .	24
<b>4</b>	<b>Sistema de acompanhamento</b>	<b>31</b>
4.1	Descrição geral . . . . .	31
4.2	Modelagem do movimento do alvo . . . . .	40
4.3	Ferramentas utilizadas . . . . .	42
4.4	Metodologia de avaliação de desempenho . . . . .	43
4.5	Resultados e discussão . . . . .	45
4.6	Deficiências do sistema e dificuldades encontradas . . . . .	49
4.7	Desenvolvimentos futuros . . . . .	52
<b>5</b>	<b>Conclusão</b>	<b>53</b>





## Lista de Figuras

2.1	Sistema de cores RGB. Adaptada de [3]. . . . .	4
2.2	Sistema de cores HSV. Adaptada de [3]. . . . .	5
2.3	Exemplo de separação das componentes H, S e V de uma imagem.	7
2.4	(a) a mão é o alvo, com base no qual o histograma-modelo foi construído; (b) imagem de teste, submetida à projeção reversa de histograma; (c) distribuição de probabilidade de cor resultante. . .	9
2.5	Ilustração do funcionamento iterativo do algoritmo <i>mean shift</i> . (a) posicionamento da janela inicial; (b) é calculado o centroide dos pontos contidos na janela; (c) posicionamento do centroide; (d) a janela é deslocada, de forma que seu centro coincida com o centroide calculado; (e) após algumas iterações, o algoritmo atinge a convergência; (f) a região com maior densidade de pontos foi encontrada. . . . .	11
2.6	Ilustração do funcionamento iterativo do algoritmo <i>mean shift</i> em uma distribuição de probabilidade de cor. (a) posicionamento da janela inicial; (b) é calculado o centroide dos pontos contidos na janela; (c) a janela é deslocada, de forma que seu centro coincida com o centroide calculado; (d) após algumas iterações, a janela se aproxima da região de maior intensidade; (e) o centroide no qual a convergência é atingida é determinado; (f) deslocamento da janela da posição inicial até a posição de convergência do algoritmo. . .	13
2.7	Diagrama esquemático das etapas do <i>camshift</i> e de outras atividades associadas. . . . .	15
2.8	Elipse que delimita o alvo e seus parâmetros, conforme produzidos pelo <i>camshift</i> . . . . .	16
2.9	Ilustração do funcionamento iterativo do algoritmo <i>camshift</i> em uma distribuição de probabilidade de cor. (a) posicionamento da janela inicial; (b) a janela é deslocada, de forma que seu centro coincida com o centroide calculado; (c) o tamanho da janela é ajustado e a elipse é definida; (d) após algumas iterações, a janela se aproxima da região de maior intensidade; (e) a janela é novamente ajustada e, a elipse, calculada; (f) comparação entre a janela inicial e a elipse obtida após a convergência. . . . .	17
3.1	Estimativa filtrada. . . . .	20
3.2	Estimativa preditora. . . . .	20
3.3	Estimativa suavizadora. . . . .	20
4.1	Exemplo de seleção de alvo. A área selecionada é dada pela porção da imagem com cores negativas. . . . .	32
4.2	Seletores para a faixa de valores de saturação (S) e valor (V) considerada para a produção do histograma-modelo. . . . .	32
4.3	O alvo (bola vermelha) e seu histograma-modelo. . . . .	33
4.4	O alvo (bola roxa) e seu histograma-modelo. . . . .	33

4.5	Abaixo, a distribuição de probabilidade de cor para a cena acima, construída com base no histograma-modelo montado para a bola vermelha. . . . .	34
4.6	Quadro típico do rastreamento do alvo. São visíveis a elipse que o contorna e o ponto vermelho que indica seu centro. No canto inferior esquerdo, a legenda em cor vermelha que indica as coordenadas desse centro. . . . .	35
4.7	Quadro de rastreamento com os preditores de posição ativos. No canto inferior esquerdo, as legendas nas cores azul e verde, que indicam as predições nominal e robusta, respectivamente. . . . .	36
4.8	Diagrama esquemático com as principais atividades do sistema de acompanhamento. . . . .	37
4.9	Gráfico atualizado em tempo real com os erros de predição de cada estimador, em pixels. . . . .	38
4.10	Gráfico atualizado em tempo real com o tempo de processamento de quadro (ms). . . . .	39
4.11	Sistema de acompanhamento operando sobre o vídeo utilizado para a avaliação de desempenho. . . . .	44
4.12	Erros de predição para os estimadores nominal e robusto, em pixels. Cada ponto da curva representa a média sobre uma série de 100 execuções da avaliação. . . . .	46
4.13	Tempo de processamento de quadro (ms) sem nenhum preditor ativo. Cada ponto da curva representa a média sobre uma série de 100 execuções da avaliação. . . . .	47
4.14	Tempo de processamento de quadro (ms) com somente o preditor nominal ativo. Cada ponto da curva representa a média sobre uma série de 100 execuções da avaliação. . . . .	48
4.15	Tempo de processamento de quadro (ms) com somente o preditor robusto ativo. Cada ponto da curva representa a média sobre uma série de 100 execuções da avaliação. . . . .	49
4.16	Situação em que o alvo (bola vermelha) se confunde com um objeto de cor semelhante ao fundo. À direita, a distribuição de probabilidade de cor do quadro. . . . .	50
4.17	Após a calibração das faixas de S e V consideradas, o alvo não mais se confunde com o objeto ao fundo. À direita, a distribuição de probabilidade de cor do quadro. . . . .	50
4.18	Predições (círculos azul e verde) atrasadas em relação à posição real do alvo (círculo vermelho). . . . .	51

## Lista de Símbolos

$I_n$	matriz identidade de dimensão $n$
$\text{posto}(A)$	posto da matriz $A$
$A^\dagger$	pseudo-inversa da matriz $A$
$A^{-1}$	inversa da matriz $A$
$A^T$	transposta da matriz $A$
$A \succeq 0$	$A$ é uma matriz semidefinida positiva
$A \succ 0$	$A$ é uma matriz definida positiva
$\ x\ $	norma Euclidiana de $x$ definida por $(x^T x)^{\frac{1}{2}}$
$\ x\ _P$	norma ponderada de $x$ definida por $(x^T P x)^{\frac{1}{2}}$ com $P \succ 0$



## 1 Introdução

São abundantes as possíveis aplicações do acompanhamento de alvos móveis em vídeos: mecanismos de interação homem-máquina, vigilância e segurança, comunicação e compressão, realidade aumentada, controle de tráfego, entre outras. Dadas a grande quantidade de dados contida no vídeo e a complexidade das sucessivas cenas e da interação do alvo com as mesmas, o rastreamento é, em geral, uma atividade desafiadora e demandante em termos de tempo computacional. Por isso mesmo, tem sido frequente a publicação de trabalhos que desenvolvem ou adaptam diferentes estratégias a instâncias específicas do problema.

Em geral, sistemas de acompanhamento de alvos possuem dois componentes principais: representação e localização do alvo, e filtragem e associação de dados. O primeiro está relacionado à identificação do objeto que se move na cena, a cada quadro. Para isso, frequentemente faz-se uso de técnicas de processamento de imagens, que incluem transformações e segmentação. Muitas das técnicas se baseiam em características como cor e geometria do alvo para localizá-lo na cena [1], ou em uma combinação de mais de uma delas [8].

Estimadores de estado, como filtros e preditores, são úteis de diferentes formas ao processo de acompanhamento do alvo, e têm sido empregados em diversos trabalhos [7] [11] [6]. Através da incorporação de informações medidas da cena ou do objeto, e da utilização de um modelo dinâmico subjacente para o movimento do mesmo, os estimadores permitem que se eleve a robustez do rastreamento em relação a inconvenientes e a ruídos em geral, como oclusões do alvo e vibrações da câmera. Além disso, preditores podem servir para reduzir a janela de processamento do quadro subsequente, aumentando a eficiência do processo.

Este trabalho de conclusão de curso apresenta um sistema de acompanhamento de trajetória de um alvo móvel, o qual agrega segmentação e rastreamento baseados no algoritmo *camshift* [3] [12] [4], derivado do conhecido *mean shift* [10], ao preditor robusto desenvolvido em [16] e [5]. Trata-se de um algoritmo preditor eficiente e especialmente adequado a aplicações de tempo real. A abordagem adotada assemelha-se à dos trabalhos [15], [18] e [14], nos quais é apresentado um sistema de rastreamento de dedos.

Permitindo a seleção, pelo usuário, do objeto a ser rastreado, o programa apresenta, em tempo real, a avaliação do desempenho do estimador robusto quanto às previsões produzidas, comparando-a àquela de um estimador nominal similar ao tradicional Filtro de Kalman. Ao mesmo tempo, avalia-se o impacto provocado no desempenho computacional pelo cômputo das estimativas a cada quadro processado.

### **1.1 Organização do texto**

O texto se inicia com a apresentação de tópicos preliminares, relacionados aos conceitos e algoritmos de processamento de imagens integrantes do componente de representação e localização do alvo. A seguir, são dadas as formulações dos estimadores de estado nominal e robusto. Neste ponto, é descrito o sistema resultante da integração dos dois componentes, com seu funcionamento e sua interface com o usuário. Por fim, são mostrados os resultados obtidos nas avaliações de desempenho conduzidas experimentalmente.

## 2 Representação e localização do alvo

O componente de representação e localização do alvo, parte fundamental do sistema de acompanhamento, tem seus fundamentos apresentados nesta seção. Por ser o rastreamento baseado em cor, faz-se inicialmente uma discussão sobre o espaço de cores HSV e suas vantagens para o processo de segmentação. Segue-se, então, com a apresentação dos algoritmos empregados para o rastreamento propriamente dito: a projeção reversa de histograma (*histogram backprojection*) e o *camshift*. O primeiro é utilizado para produzir a distribuição de probabilidade de cor, que indica, para cada pixel, a probabilidade de que o mesmo pertença à região representada por um histograma-modelo. O segundo é responsável por identificar o objeto na cena, com base na distribuição, e também a área que o mesmo ocupa. O *camshift* é um aperfeiçoamento útil do conhecido algoritmo *mean shift*, e este último, por isso, também recebe uma breve introdução.

### 2.1 O espaço de cores HSV

O algoritmo *camshift*, empregado neste trabalho para rastrear objetos com base na cor, não opera diretamente sobre os quadros capturados do vídeo, mas sobre quadros que representam distribuições de probabilidade de cor. O cômputo de tais distribuições se faz com a utilização do espaço de cores HSV (*Hue, Saturation, Value*), mais conveniente do que o tradicional RGB.

A extração de informações de uma imagem é, em uma ampla variedade de situações, bastante simplificada pela utilização da cor, que é um poderoso descritor. Formas diversas para a representação das cores foram propostas, os denominados *espaços de cores* (vide, por exemplo, [13]); cada um dos espaços pode ser mais adequado a um segmento de aplicações.

Um dos mais conhecidos espaços de cores é o RGB. Nele,

uma representação linear da cor é construída, com base em um sistema de três eixos relacionados às cores primárias - vermelho (R), verde (G) e azul (B) - como ilustra a figura 2.1. Uma cor específica é representada como um ponto no sistema de coordenadas, o qual indica a proporção de cada cor primária em sua composição. São notáveis os pontos que representam o branco, composto pela máxima quantidade de todas as cores primárias, e o preto, caracterizado pela ausência total das mesmas.

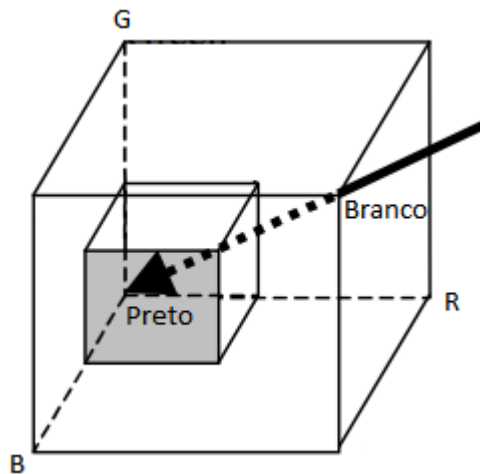


Figura 2.1: Sistema de cores RGB. Adaptada de [3].

Para aplicações que envolvem identificação e rastreamento de objetos, a segmentação de cores é realizada mais convenientemente convertendo-se os quadros para o espaço HSV (ver, por exemplo, [3] e [15]). Esse sistema corresponde à projeção do espaço RGB ao longo de sua diagonal principal, do branco ao preto, como indica a seta na figura 2.1. O resultado é o cone mostrado na figura 2.2, que possui o eixo vertical V (*Value*, ou valor), o eixo horizontal S (*Saturation*, ou saturação) e o eixo angular H (*Hue*, ou matiz).

A grande vantagem do sistema HSV advém do fato de que nele se encontram separadas as componentes de cromaticidade e



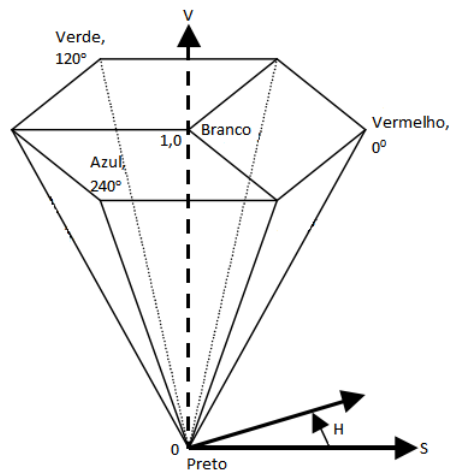


Figura 2.2: Sistema de cores HSV. Adaptada de [3].

de luminância da cor. O matiz (H) especifica particularmente o tipo da cor, enquanto a saturação (S) determina quão concentrada a cor é (o que se relaciona à quantidade de luz branca misturada com a cor pura) e o valor (V) informa a intensidade de luz (ou o brilho). O matiz da cor é afetado com menor intensidade por variações de iluminação e outros efeitos, de forma que, considerando-se uma pequena faixa de H que represente a cor do objeto, é possível manter um rastreamento mais eficaz do mesmo.

De posse dos valores R, G e B de uma cor, compreendidos em uma escala que varia de 0 a 255, a conversão para o espaço HSV pode ser realizada através das seguintes relações:

$$\begin{aligned}
R' &= \frac{R}{255} \\
G' &= \frac{G}{255} \\
B' &= \frac{B}{255} \\
C_{max} &= \max\{R', G', B'\} \\
C_{min} &= \min\{R', G', B'\} \\
\Delta &= C_{max} - C_{min} \\
H &= \begin{cases} 60^\circ \times \left( \frac{G' - B'}{\Delta} \text{mod} 6 \right), & \text{se } C_{max} = R' \\ 60^\circ \times \left( \frac{B' - R'}{\Delta} + 2 \right), & \text{se } C_{max} = G' \\ 60^\circ \times \left( \frac{R' - G'}{\Delta} + 4 \right), & \text{se } C_{max} = B' \end{cases} \quad (2.1) \\
S &= \begin{cases} 0, & \text{se } \Delta = 0 \\ \frac{\Delta}{C_{max}}, & \text{se } \Delta \neq 0 \end{cases} \quad (2.2) \\
V &= C_{max} \quad (2.3)
\end{aligned}$$

Um exemplo de separação das componentes H, S e V de uma imagem é mostrado na figura 2.3. É possível observar que regiões com cores semelhantes apresentam intensidade praticamente constante da componente H, e que cores mais puras, como o roxo da pluma do chapéu, possuem maiores intensidades da componente S. A componente V, relacionada à luminância da imagem, é a que carrega a maior parcela de informação acerca da cena registrada.



Figura 2.3: Exemplo de separação das componentes H, S e V de uma imagem.

## 2.2 Distribuição de probabilidade de cor

A operação do algoritmo *camshift*, que rastreia o objeto de interesse na cena, se dá sobre uma imagem que representa uma distribuição de probabilidade de cor, a qual é construída com base no quadro corrente do vídeo. Nessa imagem, quando em escala de cinza, a intensidade de iluminação de cada pixel é tão maior quanto maior for a probabilidade de que seu correspondente no quadro pertença ao objeto rastreado, tendo-se a cor como medida de proximidade.

Cada um dos quadros do vídeo é convertido a uma distribuição de probabilidade de cor, fazendo-se uso, para tanto, do histograma-modelo construído para os valores de H amostrados na região de rastreo selecionada. Na representação, as probabilidades se estendem em passos discretos entre o valor 0 (cor preta), que corresponde à probabilidade nula de que o pixel pertença à região, e o valor 255 (cor branca), que denota

a probabilidade unitária, ou total.

A construção da distribuição de probabilidade de cor dá-se com o uso do algoritmo de projeção reversa do histograma, ou *histogram backprojection*. Os passos são bastante simples:

1. Para cada pixel  $p_q(x,y)$  do quadro corrente, determina-se a classe do histograma-modelo à qual o valor H do mesmo pertence ( $h_{x,y}$ );
2. toma-se, do histograma-modelo, a frequência  $f_{x,y}$  correspondente à classe  $h_{x,y}$ ;
3. atribui-se, ao correspondente pixel  $p_d(x,y)$  da imagem de distribuição de probabilidade, o valor  $f_{x,y}$ ;
4. realizados os passos anteriores para cada pixel, normaliza-se a imagem de distribuição de probabilidade, para que o máximo valor corresponda a 255.

Ao final da execução do algoritmo, os pixels da imagem de distribuição de probabilidade representarão, através de maior ou menor intensidade, a probabilidade de que pertençam ao objeto rastreado, baseando-se no histograma-modelo. Para a correta operação do *camshift*, essa imagem é, em seguida, limiarizada, sendo convertida a uma imagem binária, na qual os pixels brancos apresentam grande probabilidade de serem parte do objeto rastreado.

A figura 2.4 mostra a distribuição de probabilidade de cor resultante para um caso particular de alvo e imagem de teste.

### 2.3 O algoritmo *mean shift*

Proposto em 1975 [10], o *mean shift* é um procedimento não-paramétrico para a estimação de funções de densidade de probabilidade amplamente aplicado a problemas de reconhecimento

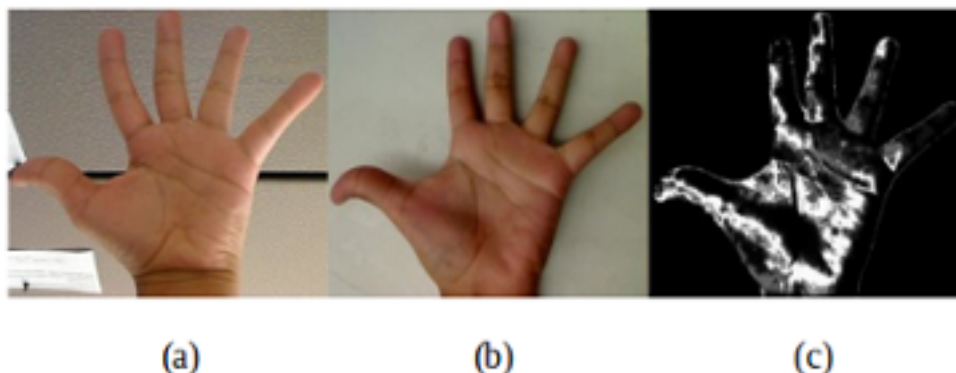


Figura 2.4: (a) a mão é o alvo, com base no qual o histograma-modelo foi construído; (b) imagem de teste, submetida à projeção reversa de histograma; (c) distribuição de probabilidade de cor resultante.

de padrões. Sua eficácia no desempenho de tarefas de visualização de baixo nível, como segmentação e rastreamento, tem sido explorada em uma grande quantidade de trabalhos [9].

Para segmentar a imagem em regiões de cor semelhante, o *mean shift* estima, para cada pixel, o gradiente da densidade local de pixels com cores semelhantes. As estimativas são utilizadas, através de um procedimento iterativo, para encontrar picos de densidade local, e todos os pixels que conduzem ao mesmo pico são tomados como parte do mesmo segmento.

Antes que o procedimento propriamente dito tenha início, deve-se definir, para cada pixel  $p_i$ , sua janela de influência, denominada *kernel*, ou janela de busca [9]. O *kernel* define uma medida intuitiva da distância entre pixels, tanto em termos espaciais quanto em termos de cor; seu tamanho e sua forma podem ter impacto significativo na eficácia da segmentação. Atribui-se então, a cada pixel, um ponto  $M(p_i)$ , que é inicializado coincidindo com o pixel. Os pontos  $M(p_i)$  são iterativamente movidos no sentido do gradiente da função de densidade definida pela soma de todos os *kernels*, até que eles atinjam um ponto estacionário, na região de máxima concentração da distribuição. Um mesmo segmento é formado por

todos os pixels associados a pontos  $M(p_i)$  que migram, aproximadamente, para o mesmo ponto estacionário. Segmentos adjacentes podem ser combinados, subsequentemente.

O algoritmo *camshift*, utilizado neste trabalho conforme apresentado em [3], emprega o *mean shift* como um de seus passos, utilizando um *kernel* quadrado. O objetivo não é segmentar a imagem em regiões de cor semelhante, mas sim determinar a região de máxima concentração de pixels com alta intensidade na distribuição de probabilidade de cor de um dado quadro. Nesse caso, o algoritmo *mean shift* é definido segundo os seguintes passos:

1. Escolhe-se o tamanho do lado da janela  $W$  de busca,  $s$ ;
2. centraliza-se a janela inicial no pixel  $p_i$ ;
3. calcula-se o centroide da janela, como

$$M(p_i) = \frac{1}{|W|} \sum_{j \in W} p_j$$

4. centraliza-se a janela no ponto  $M(p_i)$ ;
5. repetem-se os passos 3 e 4 até a convergência, quando a janela se move menos do que um determinado limiar.

Sendo  $f(p)$  a função densidade de probabilidade de cor dos pixels  $p_i$ , a qual é desconhecida, a cada passo temos

$$M(p_i) - p_i \approx \frac{f'(p_i)}{f(p_i)},$$

de forma que os  $M(p_i)$  seguem no sentido ascendente do gradiente de  $f(p)$ . Próximo à região de máxima concentração, tem-se  $f'(p) \approx 0$ , o que acarreta a convergência do algoritmo.

A figura 2.5 mostra como o *mean shift* se aproxima, iterativamente, da região de maior densidade da distribuição de

pontos. A circunferência pontilhada indica os limites da janela considerada. No quadro (b), os pontos contidos na janela são utilizados para determinar seu centroide, mostrado no quadro (c); a janela é então centralizada nesse centroide, como mostra o quadro (d). Após algumas iterações, tem-se a situação exibida no quadro (e), no qual o algoritmo atingiu a convergência. No quadro (f), a região com maior densidade de pontos foi encontrada.

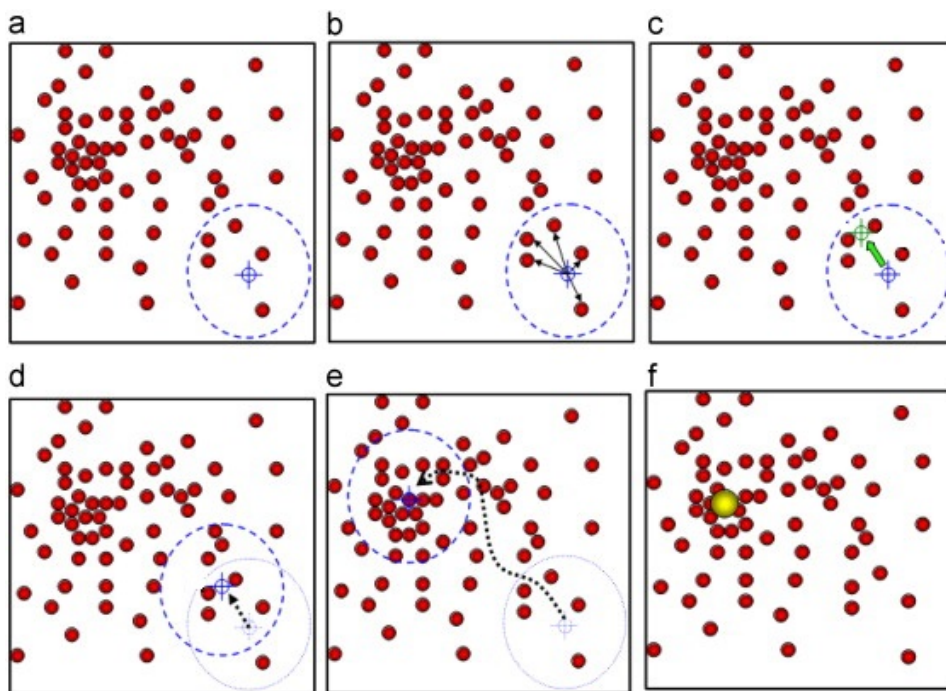


Figura 2.5: Ilustração do funcionamento iterativo do algoritmo *mean shift*. (a) posicionamento da janela inicial; (b) é calculado o centroide dos pontos contidos na janela; (c) posicionamento do centroide; (d) a janela é deslocada, de forma que seu centro coincida com o centroide calculado; (e) após algumas iterações, o algoritmo atinge a convergência; (f) a região com maior densidade de pontos foi encontrada.

Os cálculos envolvidos no algoritmo podem ser simplificados com o auxílio dos momentos da imagem de distribuição de probabilidade de cor. Sendo  $I(x,y)$  o valor de intensidade da imagem no pixel da posição  $(x,y)$ , calcula-se o momento zero

como

$$M_{00} = \sum_{x \in W} \sum_{y \in W} I(x, y)$$

Os primeiros momentos para  $x$  e para  $y$  são dados, respectivamente, por

$$M_{10} = \sum_{x \in W} \sum_{y \in W} xI(x, y)$$

$$M_{01} = \sum_{x \in W} \sum_{y \in W} yI(x, y)$$

Então, as coordenadas  $(x_c, y_c)$  do centroide  $M(p_i)$  serão

$$x_c = \frac{M_{10}}{M_{00}}$$

$$y_c = \frac{M_{01}}{M_{00}}$$

Como já foi afirmado, quando empregado pelo *camshift* no sistema de rastreamento, o *mean shift* opera sobre uma imagem que representa uma distribuição de probabilidade de cor, e seu objetivo é determinar a região da imagem com a maior probabilidade de pertencer ao alvo rastreado. O funcionamento do algoritmo em tal situação está representado na sequência de quadros da figura 2.6, na qual o alvo possui cor de pele. Na distribuição de probabilidade, os pixels com tonalidade mais próxima à da pele aparecem com maior intensidade. É possível observar que, após a convergência, o algoritmo encontra a região da imagem com a maior concentração de pixels de alta intensidade.

## 2.4 O algoritmo *camshift*

O *camshift* - *Continuously Adaptive Mean Shift*, ou *Mean Shift* Continuamente Adaptativo - algoritmo proposto em [3], surgiu como uma adaptação do *mean shift* a distribuições dinâmicas de probabilidade. Ajustando o tamanho da janela de busca no



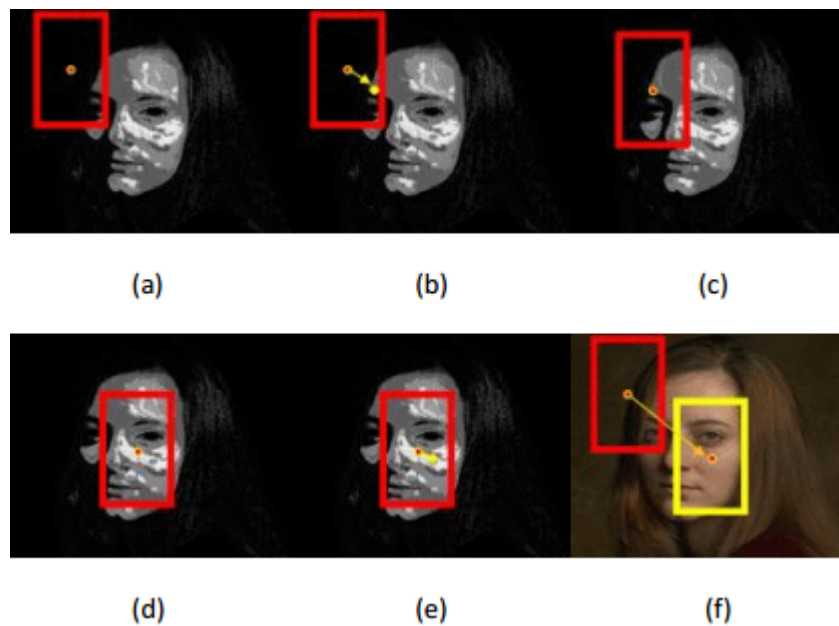


Figura 2.6: Ilustração do funcionamento iterativo do algoritmo *mean shift* em uma distribuição de probabilidade de cor. (a) posicionamento da janela inicial; (b) é calculado o centroide dos pontos contidos na janela; (c) a janela é deslocada, de forma que seu centro coincida com o centroide calculado; (d) após algumas iterações, a janela se aproxima da região de maior intensidade; (e) o centroide no qual a convergência é atingida é determinado; (f) deslocamento da janela da posição inicial até a posição de convergência do algoritmo.

curso de sua operação, o *camshift* é mais adequado ao rastreamento de objetos em vídeos, quando o objeto se move e se alteram no tempo a localização e o tamanho da distribuição de probabilidade. A adaptação do tamanho da janela baseia-se no valor do momento zero da distribuição,  $M_{00}$ , que fornece uma medida da área da mesma. A cada quadro, é produzida uma elipse, sobreposta à cena, que objetiva delimitar o alvo. O algoritmo, conforme [3], é constituído pela sequência de passos a seguir:

1. Escolhe-se uma posição inicial para a janela de busca;
2. executa-se o algoritmo *mean shift*, conforme apresentado anteriormente; armazena-se  $M_{00}$ ;

3. toma-se como tamanho da janela de busca uma função de  $M_{00}$ ,  $s(M_{00})$ ;
4. repetem-se os passos 2 e 3 até a convergência, quando a localização da janela se move menos do que um certo limiar;
5. são calculados o comprimento do eixo menor da elipse que encerra o alvo,  $w$ , o comprimento do eixo maior,  $l$ , e o ângulo deste com a horizontal,  $\theta$ .

Na figura 2.7 são esquematizadas as etapas do algoritmo *camshift*, juntamente a outras atividades associadas, como o cálculo do histograma-modelo e a produção da distribuição de probabilidade de cor.

A cada quadro, o algoritmo utiliza inicialmente a posição central e o tamanho da janela obtidos no quadro anterior. O tamanho da janela de busca, em uma imagem que, como a distribuição de probabilidade de cor, apresenta pixels com máximo valor de 255, é dado por

$$s(M_{00}) = 2\sqrt{\frac{M_{00}}{256}} \quad (2.4)$$

O cálculo dos parâmetros que definem a elipse que envolve o alvo se utiliza dos segundos momentos da imagem de distribuição de probabilidade de cor, que são os seguintes:

$$M_{20} = \sum_{x \in W} \sum_{y \in W} x^2 I(x, y)$$

$$M_{02} = \sum_{x \in W} \sum_{y \in W} y^2 I(x, y)$$

$$M_{11} = \sum_{x \in W} \sum_{y \in W} xy I(x, y)$$

São definidos os seguintes parâmetros auxiliares:

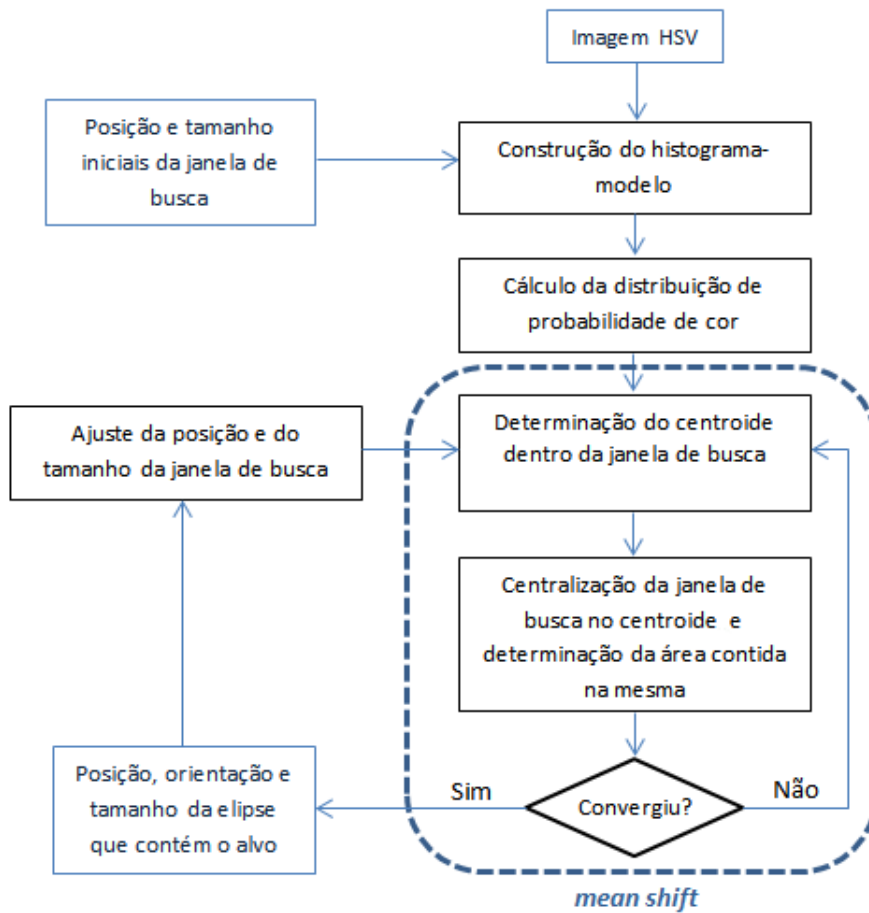


Figura 2.7: Diagrama esquemático das etapas do *camshift* e de outras atividades associadas.

$$a = \frac{M_{20}}{M_{00}} - x_c^2$$

$$b = \frac{M_{11}}{M_{00}} - x_c y_c$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2$$

Dados os parâmetros, o comprimento do eixo menor da elipse, o comprimento de seu eixo maior e o ângulo deste com a horizontal, ilustrados na figura 2.8, são calculados, respectivamente, pelas seguintes expressões:

$$w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \quad (2.5)$$

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}} \quad (2.6)$$

$$\theta = \frac{1}{2} \operatorname{tg}^{-1} \left( \frac{2 \left( \frac{M_{11}}{M_{00}} - x_c y_c \right)}{\left( \frac{M_{20}}{M_{00}} - x_c^2 \right) - \left( \frac{M_{02}}{M_{00}} - y_c^2 \right)} \right) \quad (2.7)$$

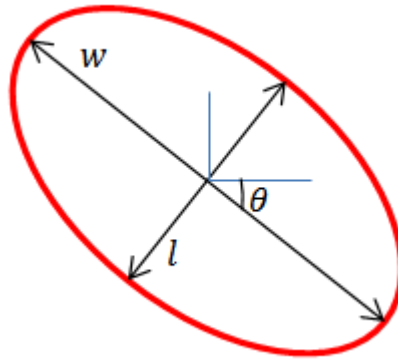


Figura 2.8: Elipse que delimita o alvo e seus parâmetros, conforme produzidos pelo *camshift*.

Está exemplificada, na figura 2.9, a aplicação do *camshift* a uma imagem que representa uma distribuição de probabilidade de cor, sendo que o alvo rastreado possui cor de pele. Como mostra a sequência de etapas, o algoritmo iterativamente atualiza a posição da janela, com o *mean shift*, ajusta seu tamanho e calcula a elipse que delimita o alvo. Após a convergência, a janela atinge a região dos pixels com maior intensidade, e a elipse abarca todo o rosto da pessoa.

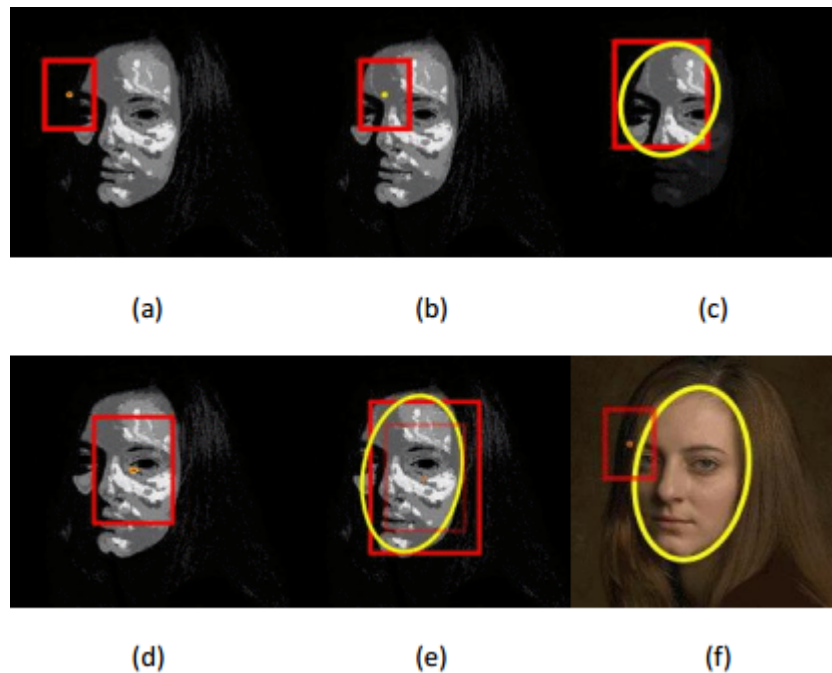


Figura 2.9: Ilustração do funcionamento iterativo do algoritmo *camshift* em uma distribuição de probabilidade de cor. (a) posicionamento da janela inicial; (b) a janela é deslocada, de forma que seu centro coincida com o centroide calculado; (c) o tamanho da janela é ajustado e a elipse é definida; (d) após algumas iterações, a janela se aproxima da região de maior intensidade; (e) a janela é novamente ajustada e, a elipse, calculada; (f) comparação entre a janela inicial e a elipse obtida após a convergência.

### 3 Filtragem e associação de dados

Esta seção é dedicada à formulação dos dois estimadores de estado utilizados como ferramenta auxiliar no processo de rastreamento do alvo. Ambos são empregados em sua forma preditora-corretora, expressa através de algoritmos recursivos que envolvem operações matriciais.

O desenvolvimento de estimadores de estado para sistemas dinâmicos é uma atividade necessária em diversas situações, como, por exemplo, no projeto de controladores em sistemas com realimentação de estado. Essa necessidade advém do fato de, por motivos físicos ou econômicos, não se dispor de senso-

res para a obtenção dos valores de cada variável de estado em cada instante de operação do sistema.

Os estimadores de estado aqui apresentados são aplicáveis a sistemas lineares de tempo discreto, representados no espaço de estados. A essência da operação de tais estimadores consiste em fornecer, para cada instante de amostragem, uma estimativa das variáveis de estado do sistema, tendo-se como base os valores das variáveis de saída do mesmo.

Para a classe de sistemas tratada neste trabalho, existe uma categoria bem conhecida de estimadores ótimos, no sentido de que otimizam uma certa função de custo, que são apresentados sob uma formulação recursiva. São os chamados Filtros de Kalman, cujo conceito fundamental advém do trabalho de Rudolf E. Kalman, na década de 1960. Desde essa concepção inicial, uma série de trabalhos tem sido conduzida, objetivando aperfeiçoar tais estimadores, já que é conhecida a perda de desempenho dos mesmos quando aplicados a sistemas com incertezas paramétricas. Como um modelo preciso de qualquer sistema é de difícil obtenção, os chamados Filtros Robustos de Kalman, que apresentam melhores propriedades quando utilizados com sistemas incertos, têm sido alvo de interesse em diferentes áreas da Engenharia.

A formulação recursiva para o filtro robusto utilizado neste trabalho, apresentada em [16] e [5], incorre em ligeiras modificações no algoritmo nominal. Ela é adequada a aplicações de tempo real que envolvam sistemas com incertezas paramétricas segundo o seguinte modelo:

$$\begin{aligned} x_{i+1} &= (F_i + \delta F_i)x_i + (G_i + \delta G_i)w_i \\ z_i &= H_i x_i + v_i \\ \begin{bmatrix} \delta F_i & \delta G_i \end{bmatrix} &= M_i \Delta_i \begin{bmatrix} E_{f,i} & E_{g,i} \end{bmatrix}, \end{aligned} \quad (3.1)$$

onde  $\{M_i, E_{f,i}, E_{g,i}\}$  são matrizes conhecidas e  $\Delta_i$  é uma matriz

de contração tal que  $\|\Delta_i\| \leq 1$ . Nesse modelo,  $w_i$  representa um ruído de modelagem de estado,  $v_i$  representa um ruído de observação,  $\delta F_i$  representa incerteza na matriz de estado  $F_i$  e  $\delta G_i$  representa incerteza na matriz  $G_i$ . A solução para o problema é baseada na resolução de um problema de mínimos quadrados regularizados com incertezas.

O filtro nominal, utilizado neste trabalho para fins de comparação com o caso robusto, considera um sistema sem incertezas como o seguinte:

$$\begin{aligned} x_{i+1} &= F_i x_i + \omega_i \\ z_i &= H_i x_i + v_i; \quad i = 0, 1, \dots \end{aligned} \quad (3.2)$$

A partir do estabelecimento de um funcional quadrático de custo e da utilização da solução do problema clássico de mínimos quadrados regularizados, obtêm-se as formulações recursivas para as estimativas ótimas para esse sistema.

Para a aplicação desenvolvida neste trabalho, é conveniente que sejam utilizadas estimativas preditoras da posição do alvo rastreado. Cabe uma breve contextualização sobre os tipos de estimativas que podem ser produzidos. Uma estimativa  $\hat{x}_{i|i}$  do estado  $x_i$  para a obtenção da qual estejam disponíveis  $i$  medidas até o instante  $i$ , *i.e.*, as medidas  $z_0, z_1, \dots, z_{i-1}, z_i$ , é denominada uma estimativa filtrada. Quando, por outro lado, dispõe-se de  $i - 1$  medidas anteriores, ou seja, das medidas  $z_0, z_1, \dots, z_{i-2}, z_{i-1}$ , a estimativa para o estado  $x_i$ ,  $\hat{x}_{i|i-1}$ , é chamada de estimativa preditora. Há também a classe das estimativas para as quais dispõe-se das medidas  $z_0, z_1, \dots, z_i, z_{i+1}$ ; nesse caso, a estimativa  $\hat{x}_{i|i+1}$  é denominada suavizadora. As figuras 3.1, 3.2 e 3.3 ilustram a obtenção das diferentes classes de estimadores.

As estimativas filtrada e preditora serão desenvolvidas para o sistema (3.2) na próxima subseção. Também será apresentada uma forma de se obter a estimativa preditora em função

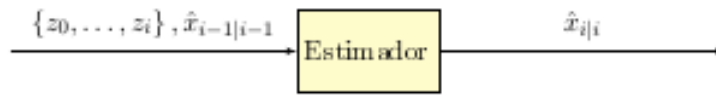


Figura 3.1: Estimativa filtrada.

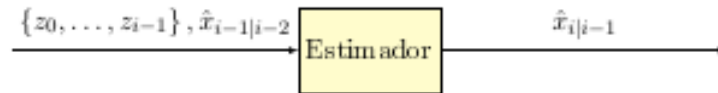


Figura 3.2: Estimativa preditora.

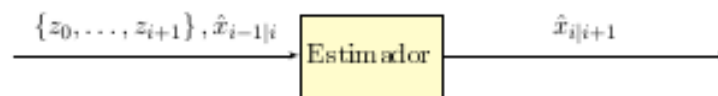


Figura 3.3: Estimativa suavizadora.

da estimativa filtrada, com poucos cálculos adicionais; essa estimativa é denominada preditora-corretora, e é a versão utilizada no sistema de acompanhamento. Posteriormente, é formulado o estimador robusto, cuja versão empregada é também a preditora-corretora.

Cumpramos observar que as deduções das estimativas aqui apresentadas seguem uma abordagem determinística baseada no método de mínimos quadrados. Uma formulação alternativa para o Filtro de Kalman, baseada em métodos estocásticos, é comum na literatura, e pode ser encontrada, por exemplo, em [2].

### 3.1 Estimador nominal

Os resultados estabelecidos a seguir, sem demonstração, fornecem as formulações para os algoritmos que produzem as estimativas filtrada, preditora e preditora-corretora para o sistema nominal. A abordagem está de acordo com aquela apresentada em [5], fundamentada nos resultados relativos à teoria de mí-



minos quadrados desenvolvida em [17] e em [16]. A solução recursiva aqui tratada servirá como base para a construção do filtro robusto.

O sistema nominal, suposto aqui, de forma genérica, como variante no tempo, é descrito pela equação de estado e pela equação de saída dadas, respectivamente, por

$$\begin{aligned} x_{i+1} &= F_i x_i + \omega_i \\ z_i &= H_i x_i + v_i; \quad i = 0, 1, \dots \end{aligned} \quad (3.3)$$

A solução do problema de filtragem consiste em encontrar uma estimativa para  $x_{i+1}$ , designada por  $\hat{x}_{i+1|i+1}$ , a partir de uma medida  $z_{i+1}$ , supondo-se conhecida a estimativa  $\hat{x}_{i|i}$  e a correspondente matriz de variância do erro de estimativa,  $P_{i|i}$ . A melhor estimativa é determinada através da solução ótima para o problema de minimização

$$J_i = \min_{x_i, x_{i+1}} \left[ \|x_i - \hat{x}_{i|i}\|_{P_{i|i}^{-1}}^2 + \|x_{i+1} - F_i x_i\|_{Q_i^{-1}}^2 + \|z_{i+1} - H_{i+1} x_{i+1}\|_{R_{i+1}^{-1}}^2 \right], \quad (3.4)$$

para  $i > 0$  e

$$J_0 = \min_{x_0} \left[ \|x_0\|_{P_0^{-1}}^2 + \|z_0 - H_0 x_0\|_{R_0^{-1}}^2 \right], \quad (3.5)$$

para  $i = 0$ .

Nas equações anteriores,  $Q_i \succ 0$  é uma matriz de ponderação para o ruído de estado,  $R_i \succ 0$  é uma matriz de ponderação para o erro de medida e  $P_0$  é a matriz de covariância do erro inicial.

A resolução pode ser obtida com base no problema de mínimos quadrados regularizados, cuja formulação é

$$\min_x \left[ x^T Q x + (Ax - b)^T W (Ax - b) \right] \quad (3.6)$$

A solução para esse problema pode ser facilmente obtida, como em [16], e é dada por

$$\hat{x} = [Q + A^T W A]^{-1} A^T W b, \quad (3.7)$$

com  $Q, W \succeq 0$  e  $Q + A^T W A$  invertível. O próximo teorema utiliza esse resultado para obter a estimativa filtrada ótima.

**Teorema 3.1.** [5] *O algoritmo recursivo abaixo produz a estimativa filtrada ótima  $\hat{x}_{i|i}$  a cada passo  $i$ :*

- *Passo 0 (condições iniciais):*

$$P_{0|0} = \left( P_0^{-1} + H_0^T R_0^{-1} H_0 \right)^{-1} \quad (3.8a)$$

$$\hat{x}_{0|0} = P_{0|0} H_0^T R_0^{-1} z_0 \quad (3.8b)$$

- *Passo  $i$ :*

$$P_{i+1|i+1} = \left( (Q_i + F_i P_{i|i} F_i^T)^{-1} + H_{i+1}^T R_{i+1}^{-1} H_{i+1} \right)^{-1} \quad (3.9a)$$

$$\hat{x}_{i+1|i+1} = F_i \hat{x}_{i|i} + P_{i+1|i+1} H_{i+1}^T R_{i+1}^{-1} (z_{i+1} - H_{i+1} F_i \hat{x}_{i|i}) \quad (3.9b)$$

A abordagem empregada para a obtenção da estimativa preditora é bastante semelhante àquela utilizada no caso da estimativa filtrada. Considerando-se novamente o sistema (3.2), agora define-se a função de custo quadrático como

$$J_i = \min_{x_i, x_{i+1}} \left[ \|x_i - \hat{x}_{i|i-1}\|_{P_{i|i-1}}^2 + \|x_{i+1} - F_i x_i\|_{Q_i}^2 + \|z_i - H_i x_i\|_{R_i}^2 \right] \quad (3.10)$$

A solução recursiva para esse problema é apresentada no próximo teorema.

**Teorema 3.2.** [5] *Supondo-se conhecida uma sequência de medidas  $z_0, z_1, \dots, z_i$ , a estimativa preditora ótima  $\hat{x}_{i+1|i}$ , segundo o critério (3.10), pode ser obtida recursivamente através do seguinte algoritmo:*

- *Passo 0 (condições iniciais):*

$$P_{0|-1} := P_0 \quad (3.11a)$$

$$\hat{x}_{0|-1} := \hat{x}_0 = 0 \quad (3.11b)$$

- *Passo  $i$ :*

$$P_{i+1|i} := Q_i + F_i P_{i|i-1} F_i^T - F_i P_{i|i-1} H_i^T (R_i + H_i P_{i|i-1} H_i^T)^{-1} H_i P_{i|i-1} F_i^T \quad (3.12a)$$

$$\hat{x}_{i+1|i} := F_i \hat{x}_{i|i-1} + F_i P_{i|i-1} H_i^T (R_i + H_i P_{i|i-1} H_i^T)^{-1} (z_i - H_i \hat{x}_{i|i-1}) \quad (3.12b)$$

As estimativas filtrada e preditora podem ser convenientemente unificadas, o que resulta em um algoritmo particularmente elegante para a obtenção desta última. A formulação é denominada preditora-corretora, e corresponde à versão implementada no sistema de acompanhamento. A estimativa preditora pode ser obtida prontamente da estimativa filtrada conforme o lema a seguir.

**Lema 3.1.** [5] *Considerando-se o modelo (3.2), a estimativa preditora  $\hat{x}_{i+1|i}$  pode ser obtida através da estimativa filtrada  $\hat{x}_{i|i}$  como*

$$\hat{x}_{i+1|i} = F_i \hat{x}_{i|i} \quad (3.13)$$

### 3.2 Estimador robusto

Seguindo uma ordem similar à da subseção anterior, aqui são apresentadas as formulações filtrada, preditora e preditora-corretora para o estimador robusto. A solução é baseada no problema de mínimos quadrados regularizados com incertezas, tratado brevemente a seguir.

É de interesse que consideremos uma função de custo quadrático com incertezas da forma

$$\|x\|_Q^2 + \|(A + \delta A)x - (b + \delta B)\|_W^2 \quad (3.14)$$

No funcional acima, a notação  $\|M\|_P$  equivale à forma quadrática ponderada  $M^T P M$ , sendo  $M$  uma matriz arbitrária e  $P$  uma matriz simétrica semi-positiva definida de ponderação. Assim,  $Q = Q^T > 0$  é uma matriz de ponderação de regularização,  $W = W^T \geq 0$  é a matriz de ponderação para o erro de estimativa,  $A$  é uma matriz  $N \times n$  conhecida e  $b$  é um vetor  $N \times 1$  conhecido. A matriz  $\delta A$  e o vetor  $\delta b$  representam, respectivamente, as perturbações da matriz nominal  $A$  e do vetor nominal  $b$ . Consideraremos a situação particular em que tais incertezas são modeladas como

$$[\delta A \ \delta b] = H\Delta[E_a \ E_b], \quad (3.15)$$

onde  $H$ ,  $E_a$  e  $E_b$  são matrizes conhecidas, e  $\Delta$  é uma matriz de contração que satisfaz a condição de que  $\|\Delta\| \leq 1$ .

O problema de otimização associado consiste em encontrar  $x$  tal que a função de custo 3.14 seja minimizada frente à máxima influência das perturbações paramétricas  $\delta A$  e  $\delta b$ , o que pode ser expresso por

$$\min_x \max_{\delta A, \delta b} \left[ \|x\|_Q^2 + \|(A + \delta A)x - (b + \delta b)\|_W^2 \right] \quad (3.16)$$

O teorema a seguir, apresentado aqui sem demonstração, fornece a solução para o problema de otimização proposto.

**Teorema 3.3.** [5], [16].

*O problema de otimização representado pelas (3.15) e (3.16) possui uma solução única  $\hat{x}$  dada por*

$$\hat{x} = (\hat{Q} + A^T \hat{W} A)^{-1} (A^T \hat{W} b + \hat{\lambda} E_a^T E_b), \quad (3.17)$$

*onde as matrizes de ponderação modificadas  $\hat{Q}$  e  $\hat{W}$  possuem a seguinte definição:*

$$\hat{Q} := Q + \hat{\lambda} E_a^T E_a \quad (3.18)$$

$$\hat{W} := W + WH (\hat{\lambda} I - H^T WH)^\dagger H^T W \quad (3.19)$$

*A representação  $M^\dagger$  denota a pseudo-inversa da matriz  $M$ . O parâmetro  $\hat{\lambda}$  é um escalar positivo determinado a partir do seguinte problema de otimização:*

$$\hat{\lambda} := \arg \max_{\lambda \geq \|H^T WH\|} G(\lambda), \quad (3.20)$$

*sendo*

$$G(\lambda) := \|x(\lambda)\|_{\hat{Q}}^2 + \lambda \|E_a x(\lambda) - E_b\|^2 + \|Ax(\lambda) - b\|_{W(\lambda)}^2, \quad (3.21)$$

*com as funções auxiliares dadas por*

$$x(\lambda) := [Q(\lambda) + A^T W(\lambda) A]^{-1} [A^T W(\lambda) b + \lambda E_a^T E_b] \quad (3.22)$$

$$Q(\lambda) := Q + \lambda E_a^T E_a \quad (3.23)$$

$$W(\lambda) := W + WH (\lambda I - H^T WH)^\dagger H^T W. \quad (3.24)$$

*Com  $x$  dado pela 3.17, a função de custo 3.14 assume seu valor ótimo, dado por*

$$G(\hat{\lambda}) = \begin{bmatrix} 1 \\ \hat{x} \end{bmatrix}^T \begin{bmatrix} \hat{\lambda} \|E_b\|^2 & -\hat{\lambda} E_b^T E_a \\ -\hat{\lambda} E_a^T E_b & Q(\lambda) \end{bmatrix} \begin{bmatrix} 1 \\ \hat{x} \end{bmatrix} + (A\hat{x} - b)^T W(\lambda) (A\hat{x} - b). \quad (3.25)$$

A formulação do estimador robusto é construída com base no sistema com incertezas paramétricas representado por

$$\begin{aligned} x_{i+1} &= (F_i + \delta F_i) x_i + (G_i + \delta G_i) w_i \\ z_i &= H_i x_i + v_i \\ \begin{bmatrix} \delta F_i & \delta G_i \end{bmatrix} &= M_i \Delta_i \begin{bmatrix} E_{f,i} & E_{g,i} \end{bmatrix} \end{aligned} \quad (3.26)$$

No modelo,  $M_i$ ,  $E_{f,i}$  e  $E_{g,i}$  são matrizes conhecidas, e  $\Delta_i$  é uma matriz de contração que satisfaz  $\|\Delta_i\| \leq 1$ . Supondo-se conhecidas as estimativas filtradas de  $x_i$ ,  $\hat{x}_{i|i}$ , e de  $w_i$ ,  $\hat{w}_{i|i} = 0$ , e obtendo-se os valores verdadeiros  $x_i$  e  $w_i$  em vista de uma nova medida  $z_{i+1}$ , podem ser definidos os seguintes erros de predição:

$$e_{x,i} := x_i - \hat{x}_{i|i} \quad (3.27)$$

$$e_{w,i} := w_i - \hat{w}_{i|i} = w_i \quad (3.28)$$

$$e_{y,i+1} := z_{i+1} - H_{i+1} x_{i+1} = v_{i+1} \quad (3.29)$$

Observa-se que o fato de que o erro de predição para a variável  $w_i$  é igual à própria  $w_i$  denota o desconhecimento de  $w_i$ , assim como o fato de o erro de predição para a saída ser igual a  $v_i$  decorre de que a representação do desconhecimento da saída recai sobre esse sinal de ruído. Os erros de predição possuem natureza aleatória.

Para o desenvolvimento da estimativa filtrada robusta ótima, define-se o seguinte funcional de custo para os erros de predição:

$$J_i := \|x_i - \hat{x}_{i|i}\|_{P_{i|i}^{-1}}^2 + \|w_i\|_{Q_i^{-1}}^2 + \|z_{i+1} - H_{i+1}x_{i+1}\|_{R_{i+1}^{-1}}^2 \quad (3.30)$$

Na função de custo acima,  $P_{i|i}$  é uma matriz de ponderação para o erro de predição de estado,  $e_{x,i}$ ,  $Q_i$  é uma matriz de ponderação para o erro de predição do ruído de estado,  $e_{w,i}$ , e  $R_{i+1}$  é uma matriz de ponderação para o erro de predição da saída. Observa-se que os dois primeiros termos de  $J_i$  representam o custo do erro de predição na ausência da informação  $z_{i+1}$ , ao passo que o terceiro termo traz a inclusão de tal atualização da saída. Assumindo-se conhecidas as estimativas filtradas  $\hat{x}_{i|i}$  e  $\hat{w}_{i|i}$ , que constituem a solução para o problema de minimização dado por

$$\min_{x_i, w_i} \left\{ \|x_i - \hat{x}_{i|i}\|_{P_{i|i}^{-1}}^2 + \|w_i\|_{Q_i^{-1}}^2 \right\}, \quad (3.31)$$

as mesmas podem ser atualizadas, em vista da nova saída aferida  $z_{i+1}$ , para as estimativas  $\hat{x}_{i|i+1}$  e  $\hat{w}_{i|i+1}$ , pela solução do seguinte problema de otimização quadrático do tipo min – max:

$$\begin{bmatrix} \hat{x}_{i|i+1} \\ \hat{w}_{i|i+1} \end{bmatrix} := \min_{\{x_i, w_i\}} \max_{\{\delta F_i, \delta G_i\}} \left\{ \|x_i - \hat{x}_{i|i}\|_{P_{i|i}^{-1}}^2 + \|w_i\|_{Q_i^{-1}}^2 + \|z_{i+1} - H_{i+1}x_{i+1}\|_{R_{i+1}^{-1}}^2 \right\}, \quad (3.32)$$

sujeito às restrições lineares

$$\begin{aligned} x_{i+1} &= (F_i + \delta F_i) + (G_i \delta G_i) w_i \\ z_{i+1} &= H_{i+1}x_{i+1} + v_{i+1} \\ \begin{bmatrix} \delta F_i & \delta G_i \end{bmatrix} &= M_i \Delta_i \begin{bmatrix} E_{f,i} & E_{g,i} \end{bmatrix}, \|\Delta_i\| \leq 1 \end{aligned}$$

Pode ser mostrado que, a partir das estimativas  $\hat{x}_{i|i+1}$  e  $\hat{w}_{i|i+1}$ , é possível obter a estimativa filtrada para  $x_{i+1}$  como

$$\hat{x}_{i+1|i+1} := F_i + G_i \hat{w}_{i|i+1} = \begin{bmatrix} F_i & G_i \end{bmatrix} \begin{bmatrix} \hat{x}_{i|i+1} \\ \hat{w}_{i|i+1} \end{bmatrix} \quad (3.33)$$

O problema (3.32) - (3.33) pode ser resolvido com o uso do teorema 3.3. Uma forma mais conveniente de solução, na forma de uma estimativa do tipo preditora-corretora, é apresentada no próximo teorema. Essa é a versão implementada no sistema de acompanhamento.

**Teorema 3.4.** [5]. *Seja o sistema linear representado pelo seguinte modelo:*

$$\begin{aligned} x_{i+1} &= (F_i + \delta F_i)x_i + (G_i + \delta G_i)w_i \\ z_i &= H_i x_i + v_i \\ \begin{bmatrix} \delta F_i & \delta G_i \end{bmatrix} &= M_i \Delta_i \begin{bmatrix} E_{f,i} & E_{g,i} \end{bmatrix}, \|\Delta_i\| \leq 1 \end{aligned} \quad (3.34)$$

*Suponham-se conhecidas as matrizes de ponderação  $\Pi_0 > 0$ ,  $R_i > 0$  e  $Q_i > 0$ . O algoritmo a seguir permite o cálculo das estimativas filtradas dos estados.*

- *Passo 0. As condições iniciais  $P_{0|0}$  e  $\hat{x}_{0|0}$  são:*

$$P_{0|0} := \left( \Pi_0^{-1} + H_0^T R_0^{-1} H_0 \right)^{-1} \quad (3.35)$$

$$\hat{x}_{0|0} := P_{0|0} H_0^T R_0^{-1} z_0 \quad (3.36)$$

- *Passo 1: Se  $H_{i+1} M_i = 0$ , então toma-se  $\hat{\lambda}_i = 0$ . Caso contrário, determina-se  $\hat{\lambda}_i$  no intervalo*

$$\lambda > \lambda_{l,i} := \|M_i^T H_{i+1}^T R_{i+1}^{-1} H_{i+1} M_i\|, \quad (3.37)$$

*de forma a minimizar a  $G(\lambda)$  definida em (3.21), com certas identificações a serem apresentadas posteriormente.*



- *Passo 2: Substituir  $Q_i$ ,  $R_{i+1}$ ,  $P_{i|i}$ ,  $G_i$  e  $F_i$  por*

$$\hat{Q}_i^{-1} = Q_i^{-1} + \hat{\lambda}_i E_{g,i}^T \left( I + \hat{\lambda}_i E_{f,i} P_{i|i} E_{f,i}^T \right)^{-1} E_{g,i} \quad (3.38)$$

$$\hat{R}_{i+1} = R_{i+1} - \hat{\lambda}_i^{-1} H_{i+1} M_i M_i^T H_{i+1}^T \quad (3.39)$$

$$\hat{P}_{i|i} = \left( P_{i|i}^{-1} + \hat{\lambda}_i E_{f,i}^T E_{f,i} \right)^{-1} \quad (3.40)$$

$$= P_{i|i} - P_{i|i} E_{f,i}^T \left( \hat{\lambda}_i^{-1} I + E_{f,i} P_{i|i} E_{f,i}^T \right)^{-1} E_{f,i} P_{i|i} \quad (3.41)$$

$$\hat{G}_i = G_i - \hat{\lambda}_i F_i \hat{P}_{i|i} E_{f,i}^T E_{g,i} \quad (3.42)$$

$$\hat{F}_i = \left( F_i - \hat{\lambda}_i \hat{G}_i \hat{Q}_i E_{g,i}^T E_{f,i} \right) \left( I - \hat{\lambda}_i \hat{P}_{i|i} E_{f,i}^T E_{f,i} \right)^{-1} \quad (3.43)$$

Se  $\hat{\lambda}_i = 0$ , as igualdades acima se reduzem a  $\hat{Q}_i = Q_i$ ,  $\hat{R}_{i+1} = R_{i+1}$ ,  $\hat{P}_{i|i} = P_{i|i}$ ,  $\hat{G}_i = G_i$  e  $\hat{F}_i = F_i$ .

- *Passo 3: A seguinte atualização é aplicada a  $\hat{x}_{i|i}$  e a  $P_{i|i}$ :*

$$\hat{x}_{i+1|i} = \hat{F}_i \hat{x}_{i|i} \quad (3.44)$$

$$\hat{x}_{i+1|i+1} = \hat{x}_{i+1|i} + P_{i+1|i+1} H_{i+1}^T \hat{R}_{i+1}^{-1} e_{i+1} \quad (3.45)$$

$$e_{i+1} = z_{i+1} - H_{i+1} \hat{x}_{i+1|i} \quad (3.46)$$

$$P_{i+1|i} = F_i \hat{P}_{i|i} F_i^T + \hat{G}_i \hat{Q}_i \hat{G}_i^T \quad (3.47)$$

$$P_{i+1|i+1} = P_{i+1|i} - P_{i+1|i} H_{i+1}^T R_{e,i+1}^{-1} H_{i+1} P_{i+1|i} \quad (3.48)$$

$$R_{e,i+1} = \hat{R}_{i+1} + H_{i+1} P_{i+1|i} H_{i+1}^T \quad (3.49)$$

Observa-se que  $\hat{x}_{i+1|i}$  é a estimativa preditora, enquanto que  $\hat{x}_{i+1|i+1}$  é a estimativa filtrada. A relação entre ambas caracteriza a estimativa preditora-corretora.

Cumpramos observar que  $\hat{Q}_i$  pode ser dada por uma expressão mais simples, com uma inversão matricial a menos, como

$$\begin{aligned}
\hat{Q}_i &:= \left( Q_i^{-1} + \hat{\lambda}_i E_{g,i}^T E_{g,i} - \hat{\lambda}_i^2 E_{g,i}^T E_{f,i} \underbrace{\left( P_{i|i}^{-1} + \hat{\lambda}_i E_{f,i}^T E_{f,i} \right)^{-1}}_{\hat{P}_{i|i}} E_{f,i}^T E_{g,i} \right)^{-1} \\
&= \left( Q_i^{-1} + \hat{\lambda}_i E_{g,i}^T \left[ I - \hat{\lambda}_i E_{g,i}^T E_{f,i} \hat{P}_{i|i} E_{f,i}^T \right] E_{g,i} \right)^{-1} \tag{3.50}
\end{aligned}$$

## 4 Sistema de acompanhamento

Esta seção destina-se à descrição do sistema resultante da integração do algoritmo *camshift* aos preditores de estado, bem como de sua interface de seleção e de acompanhamento do alvo. Também aqui serão apresentados os resultados obtidos a partir de duas avaliações de desempenho conduzidas junto ao sistema. A primeira delas possui o objetivo de validar a eficácia das predições produzidas pelos dois estimadores, comparando-as entre si. A segunda procura demonstrar a eficiência do sistema em termos de tempo computacional.

### 4.1 Descrição geral

Ao ter sua execução iniciada, o programa exibe na tela as imagens capturadas pela câmera, em tempo real. O usuário deve selecionar, com o *mouse*, o objeto-alvo. Para tanto, basta que o mesmo clique e arraste sobre este, envolvendo-o com a janela com cores invertidas que acompanha o cursor. Um exemplo de seleção de alvo é apresentado na figura 4.1.

Uma vez selecionado o alvo, o programa dá início à sequência de preparação para o rastreamento. Inicialmente, o quadro é convertido para o espaço de cor HSV. Do quadro convertido, constrói-se uma nova imagem, tomando-se os valores da componente de matiz (H) de cada pixel. É extraída dessa imagem a região correspondente à janela selecionada pelo usuário. Dessa janela, são considerados apenas os pixels que possuam os valores de saturação (S) e valor (V) compreendidos nas faixas especificadas através dos seletores localizados na parte superior da tela, visíveis na figura 4.2; isso permite uma melhor segmentação da área do objeto. O histograma-modelo, que é utilizado durante todo o processo de rastreamento, é construído para os valores de matiz (H) dessa janela limiarizada. A figura 4.3 mostra o histograma resultante, com H crescente no eixo

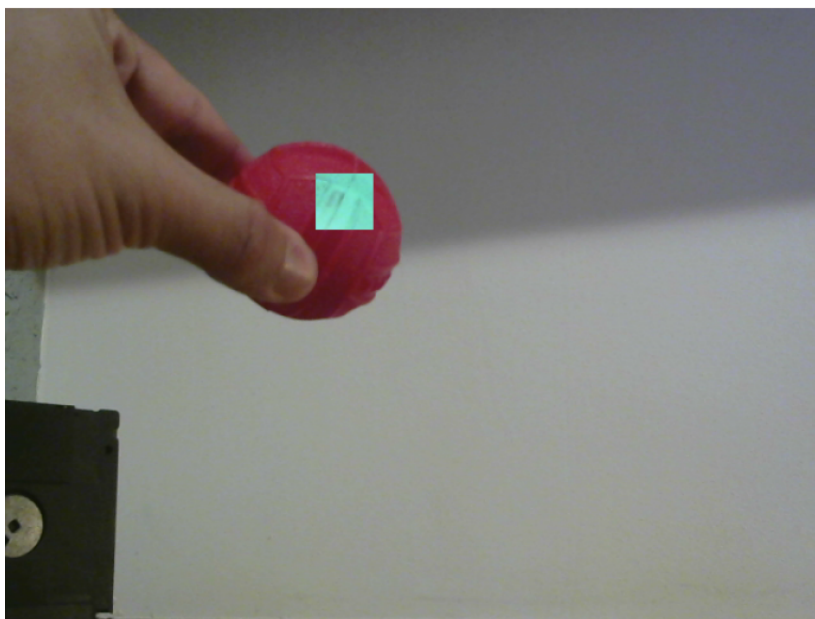


Figura 4.1: Exemplo de seleção de alvo. A área selecionada é dada pela porção da imagem com cores negativas.

horizontal. São considerados 32 intervalos de H, e as barras do histograma representam a quantidade de pixels na janela de seleção cuja componente H está compreendida em cada um dos intervalos. Na figura 4.4, há um histograma obtido com a seleção de um objeto de outra cor; é possível observar a diferença em relação ao anterior.



Figura 4.2: Seletores para a faixa de valores de saturação (S) e valor (V) considerada para a produção do histograma-modelo.

Então, passa-se a executar, para cada quadro do vídeo, o algoritmo de projeção reversa de histograma, com base no histograma-modelo. Na figura 4.5, nota-se que a imagem de distribuição de probabilidade de cor resultante visivelmente representa a silhueta do alvo.

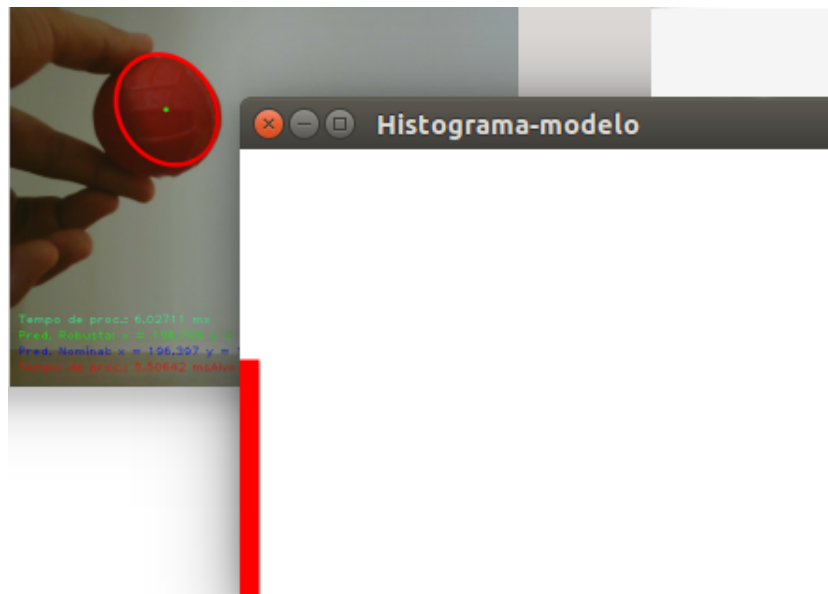


Figura 4.3: O alvo (bola vermelha) e seu histograma-modelo.

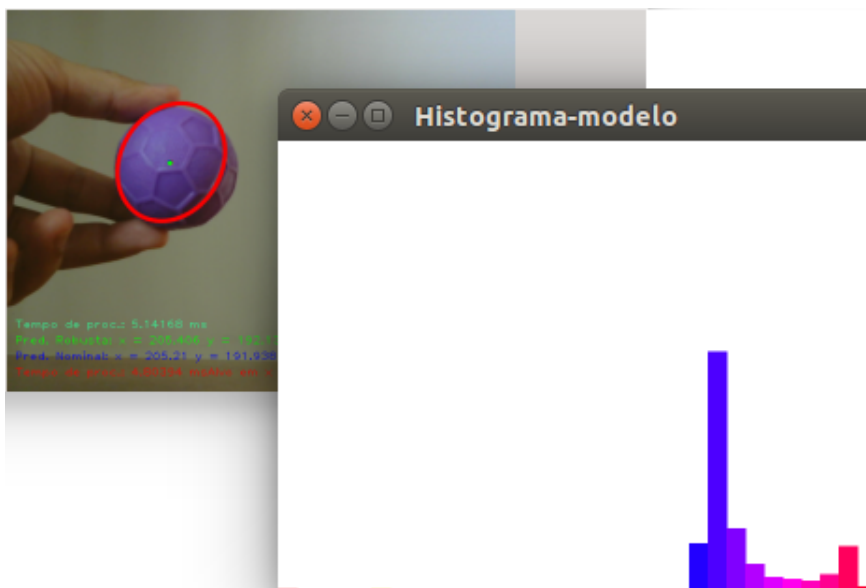


Figura 4.4: O alvo (bola roxa) e seu histograma-modelo.

Em seguida, o algoritmo *camshift*, apresentado anteriormente, é aplicado à distribuição de probabilidade de cor. Um pequeno círculo vermelho indica a posição do centro do objeto, conforme fornecida pelo algoritmo, e uma elipse, também em cor

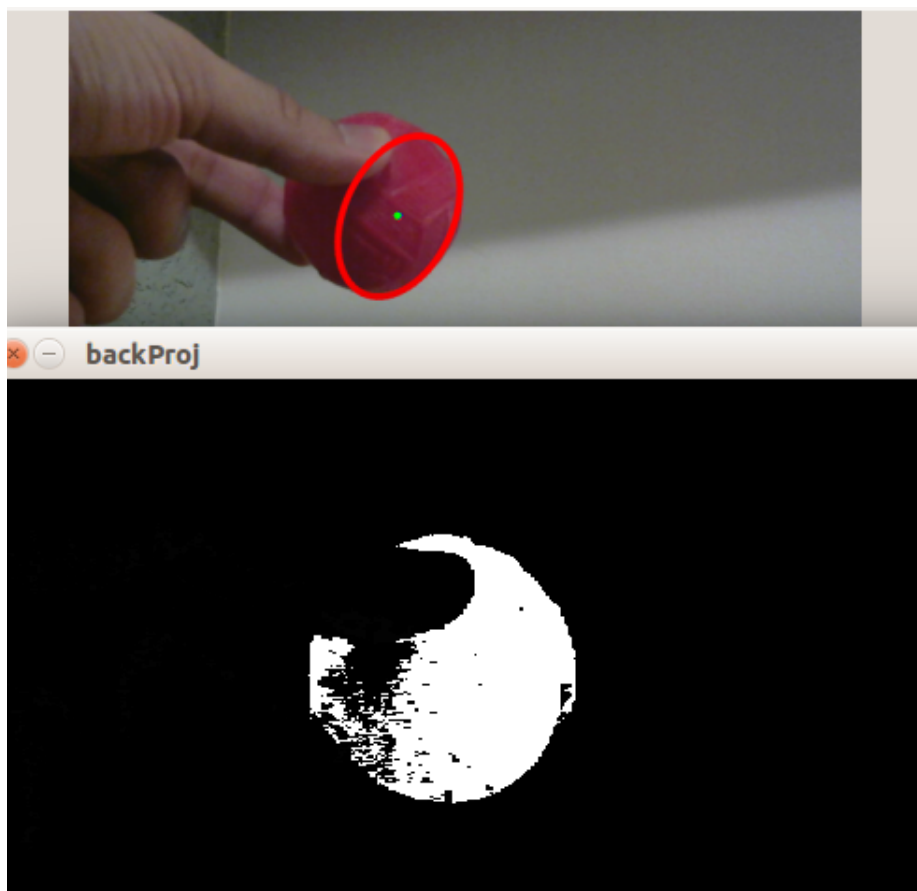


Figura 4.5: Abaixo, a distribuição de probabilidade de cor para a cena acima, construída com base no histograma-modelo montado para a bola vermelha.

vermelha, delimita a região compreendida pela área do mesmo. As coordenadas do centro calculado do objeto são dadas, de forma textual, na legenda de cor vermelha, localizada no canto inferior esquerdo da janela. Um exemplo pode ser observado na figura 4.6.

É possível, a qualquer momento, inicializar a execução dos preditores de estado nominal e robusto, através de comandos do teclado. Os mesmos comandos podem ser utilizados para encerrar a execução de cada um. As posições previstas pelos estimadores para o quadro subsequente são mostradas através de dois pequenos círculos: um na cor azul, para o preditor nomi-

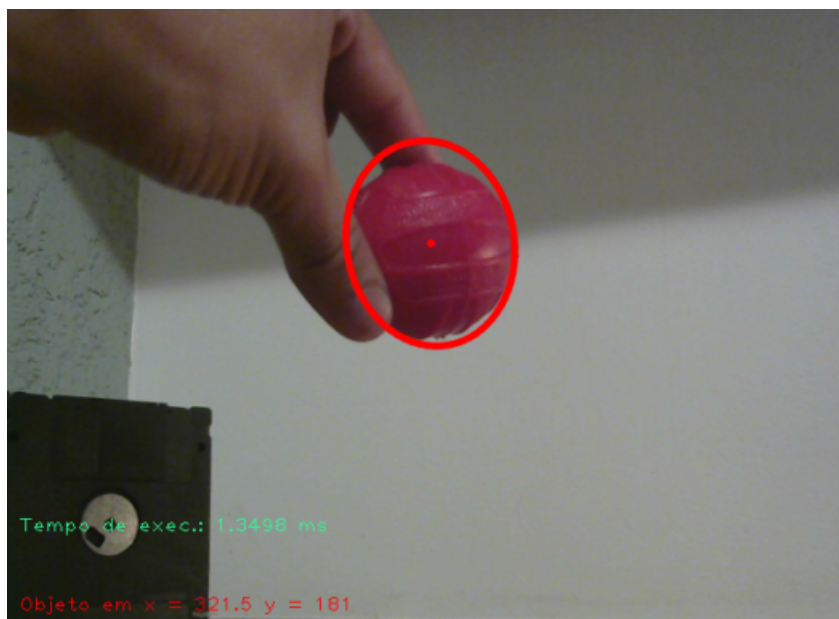


Figura 4.6: Quadro típico do rastreamento do alvo. São visíveis a elipse que o contorna e o ponto vermelho que indica seu centro. No canto inferior esquerdo, a legenda em cor vermelha que indica as coordenadas desse centro.

nal, e um na cor verde, para o preditor robusto. Além disso, as posições são especificadas por legendas, respectivamente nas mesmas cores para cada filtro, no canto inferior esquerdo da tela. A figura 4.7 mostra a tela do programa com os dois preditores em execução.

Também é visível, na figura 4.7, a indicação do tempo médio de processamento de um quadro, dado pela soma dos tempos de execução do algoritmo *camshift* e dos preditores que estiverem ativos. A média calculada é do tipo móvel exponencialmente ponderada, que pondera os valores mais antigos com pesos exponencialmente decrescentes, de forma a privilegiar os tempos de processamento dos quadros mais recentes. Sendo  $M_i$  a média para quadro  $i$  e  $t_i$  o tempo de processamento desse quadro, o cálculo é feito da seguinte forma:

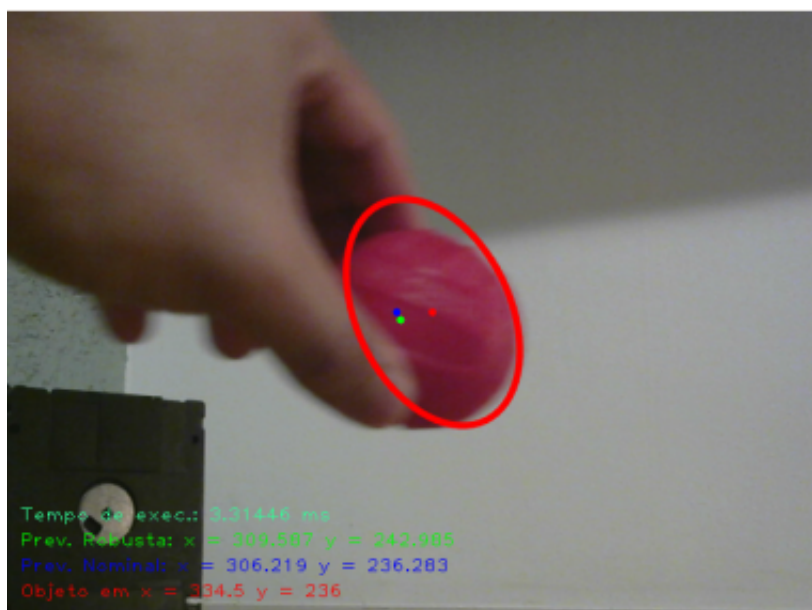


Figura 4.7: Quadro de rastreamento com os preditores de posição ativos. No canto inferior esquerdo, as legendas nas cores azul e verde, que indicam as previsões nominal e robusta, respectivamente.

$$\begin{aligned}
 M_0 &= t_0 \\
 M_i &= 0,5t_i + 0,5M_{i-1}, \text{ para } i > 0
 \end{aligned}
 \quad (4.1)$$

O diagrama da figura 4.8 ilustra as principais atividades envolvidas no sistema de acompanhamento.

Enquanto os preditores estão em execução, o programa exibe, em uma segunda janela, um gráfico, atualizado em tempo real, que mostra os erros de previsão de cada um. Um exemplo é apresentado na figura 4.9. O erro de previsão, em pixels, é definido como a distância euclidiana entre a posição do objeto no quadro corrente,  $s_i = [x_i \ y_i]^T$ , fornecida pelo *camshift*, e a posição predita no quadro anterior,  $\hat{s}_{i|i-1}$ , ou

$$E_i = ||s_i - \hat{s}_{i|i-1}|| \quad (4.2)$$

Também é exibido pelo programa o gráfico, atualizado em



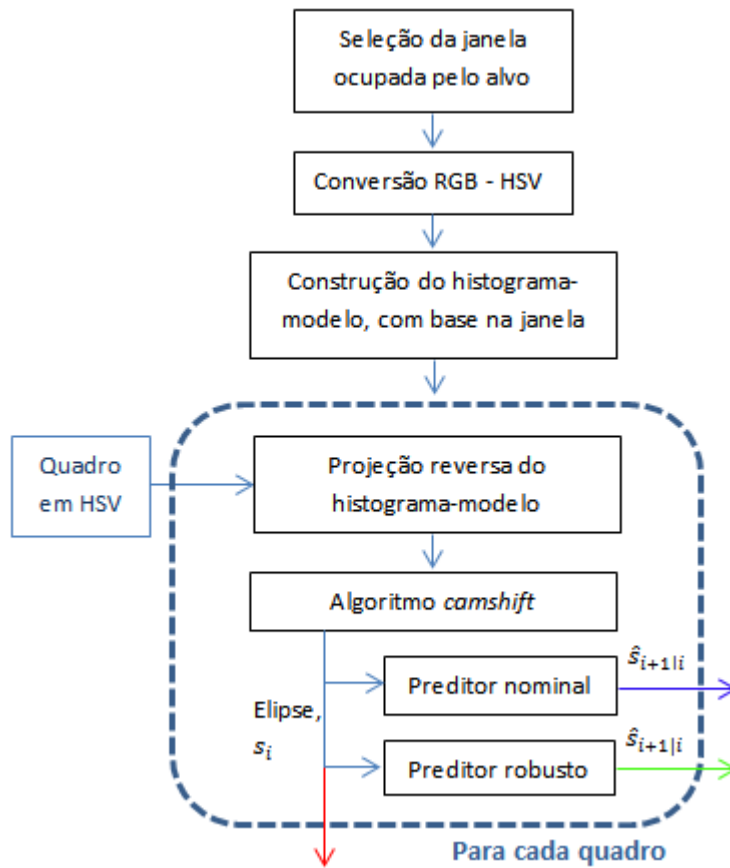
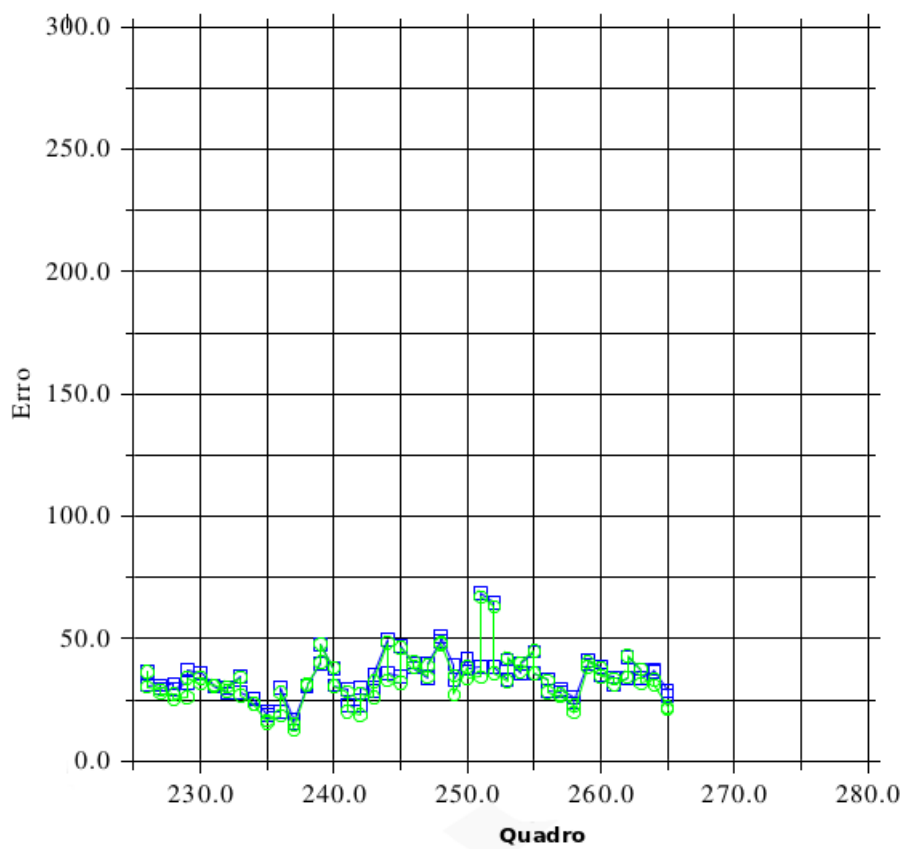


Figura 4.8: Diagrama esquemático com as principais atividades do sistema de acompanhamento.

tempo real, com o tempo de processamento de quadro, visível na figura 4.10. Ambos os gráficos podem ser salvos em arquivo através de um comando adequado do teclado. A tabela 1 sumariza os comandos de teclado aceitos pelo programa.



Media: 34.34

Media: 32.57

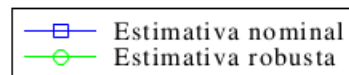
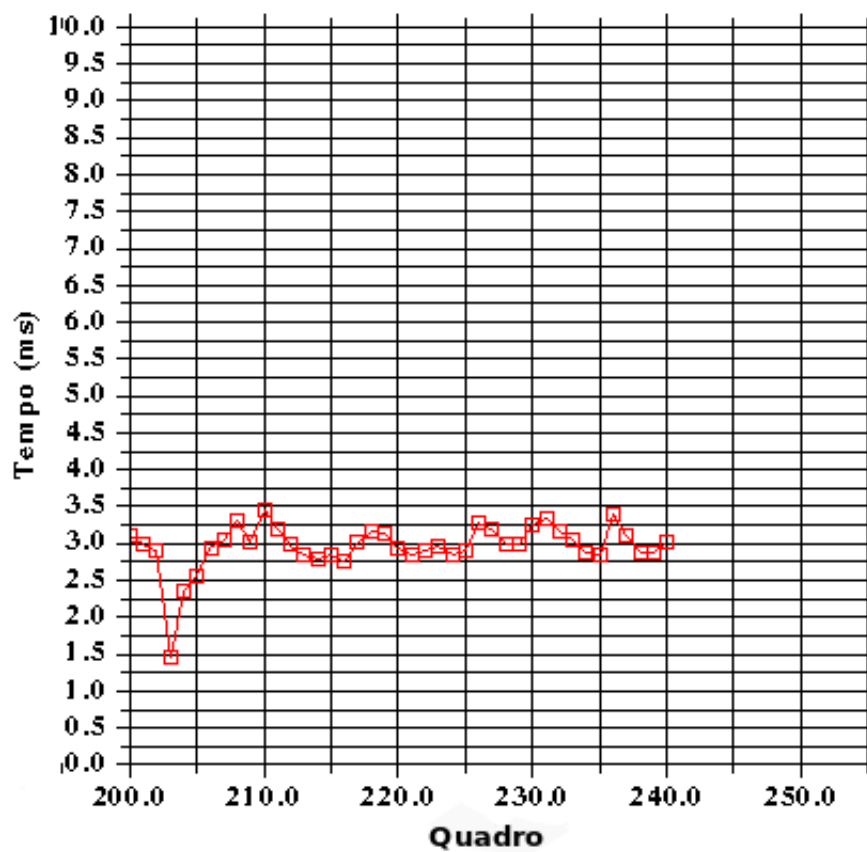


Figura 4.9: Gráfico atualizado em tempo real com os erros de predição de cada estimador, em pixels.

Tabela 1: Teclas de comando e suas respectivas funções

Tecla	Função
b	exibe a imagem de distribuição de probabilidade de cor
h	exibe o histograma-modelo
p	pausa/retoma o rastreamento do alvo
n	inicia/encerra a execução do preditor nominal
r	inicia/encerra a execução do preditor robusto
s	salva em arquivo o gráfico de erro de predição
s	salva em arquivo o gráfico de tempo de processamento
c	encerra o rastreamento do alvo



Media: 2.95

Figura 4.10: Gráfico atualizado em tempo real com o tempo de processamento de quadro (ms).

## 4.2 Modelagem do movimento do alvo

A exemplo do que foi feito em [15], [18] e [14], é utilizado um modelo linear e invariante no tempo para o movimento do alvo. Sejam  $\{x_i, y_i\}$ ,  $\{v_{x,i}, v_{y,i}\}$  (*pixels/segundo*) e  $\{\alpha_{x,i}, \alpha_{y,i}\}$  (*pixels/segundo<sup>2</sup>*) a posição, a velocidade e a aceleração do alvo no quadro corrente, respectivamente, nas direções horizontal ( $x$ ) e vertical ( $y$ ). Seja também  $T$  o intervalo entre quadros sucessivos capturados, com o valor de  $T = 1/20$  s. Então, tem-se

$$x_{i+1} \approx x_i + v_{x,i}T + \alpha_{x,i}\frac{T^2}{2} \quad (4.3)$$

$$y_{i+1} \approx y_i + v_{y,i}T + \alpha_{y,i}\frac{T^2}{2} \quad (4.4)$$

$$v_{x,i+1} \approx v_{x,i} + \alpha_{x,i}T \quad (4.5)$$

$$v_{y,i+1} \approx v_{y,i} + \alpha_{y,i}T \quad (4.6)$$

No formato do sistema livre de incertezas paramétricas 3.3, o movimento do alvo é, portanto, descrito como

$$\begin{aligned} p_{i+1} &= F p_i + G \omega_i \\ z_i &= H p_i + u_i, \end{aligned} \quad (4.7)$$

onde

$$p_i = [x_i \ y_i \ v_{x,i} \ v_{y,i} \ \alpha_{x,i} \ \alpha_{y,i}]^T \quad (4.8)$$

$$z_i = [x_i \ y_i]^T \quad (4.9)$$

e

$$F = \begin{bmatrix} 1 & 0 & T & 0 & \frac{1}{2}T^2 & 0 \\ 0 & 1 & 0 & T & 0 & \frac{1}{2}T^2 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.10)$$

$$G = I_{6 \times 6} \quad (4.11)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.12)$$

A saída do sistema,  $z_i = s_i$ , constituída pela posição  $[x_i \ y_i]^T$  do alvo no quadro corrente, é fornecida pela execução do algoritmo *camshift*. A partir dela, o estimador produz as predições da posição, da velocidade e da aceleração do alvo no quadro seguinte. O algoritmo empregado nesta situação é o da estimativa preditora-corretora nominal, apresentado nos teoremas 3.1 e 3.2 e no lema 3.1. As matrizes de ponderação  $Q$  e  $R$ , necessárias à execução do algoritmo, foram adotadas, após testes empíricos de desempenho, como sendo

$$Q = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,25 \end{bmatrix} \quad (4.13)$$

$$R = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \quad (4.14)$$

Tendo-se em vista a existência de incertezas na modelagem do sistema, faz-se necessária uma representação alternativa que dê conta de incluí-las. O sistema nos moldes daquele associado ao preditor robusto, conforme 3.26, é dado por

$$\begin{aligned}
p_{i+1} &= (F + \delta F) p_i + (G + \delta G) w_i \\
z_i &= H p_i + u_i \\
\begin{bmatrix} \delta F & \delta G \end{bmatrix} &= M \Delta \begin{bmatrix} E_f & E_g \end{bmatrix}, \tag{4.15}
\end{aligned}$$

com  $\{M, E_f, E_g\}$  convenientemente escolhidas e  $\Delta$  uma matriz de contração, tal que  $\|\Delta\| \leq 1$ . Com base nas referências [15], [18] e [14] e em testes empíricos de desempenho, os valores adotados para essas matrizes foram como abaixo. As matrizes de ponderação  $Q$  e  $R$  tiveram os valores dados em 4.13. Convém notar que o algoritmo do preditor robusto independe do valor de  $\Delta$ .

$$E_f = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{4.16}$$

$$E_g = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.17}$$

$$M = \begin{bmatrix} 0,2 & 0,2 & 0,1 & 0,1 & 0,1 & 0,1 \end{bmatrix} \tag{4.18}$$

### 4.3 Ferramentas utilizadas

O código que implementa o projeto foi construído em linguagem C++, com o apoio de bibliotecas especiais para o desempenho de certas funções. Em termos de *hardware*, além do computador que executa o programa, foi utilizada uma *webcam* para a captura das imagens.

Para as atividades de processamento de imagens em geral, contou-se com o uso do *OpenCV 2.4.9* ([www.opencv.org](http://www.opencv.org)), uma biblioteca de *software* de código aberto que compreende ferramentas de visão computacional e aprendizado de máquina. Com os recursos oferecidos por essa biblioteca, torna-se mais prática e eficiente a utilização dos algoritmos *camshift* e projeção reversa de histograma, além de operações como cálculo de histograma, limiarização de imagens, entre outras.

As operações matriciais que abundam nos algoritmos dos preditores de estado puderam ser implementadas de forma mais simplificada, e com maior eficiência, com o emprego da biblioteca *Eigen* ([eigen.tuxfamily.org](http://eigen.tuxfamily.org)), que agrega um conjunto de funções e definições para álgebra linear.

#### 4.4 Metodologia de avaliação de desempenho

Dois aspectos de desempenho do sistema foram avaliados: a qualidade das previsões produzidas pelos estimadores nominal e robusto e o impacto provocado no tempo de processamento de quadro pela execução dos algoritmos preditores. Para a condução dos testes, foi utilizada, em vez da captura em tempo real da câmera, uma sequência de vídeo previamente gravada. Nessa sequência, o alvo - uma pequena bola roxa - é movimentado de forma a completar duas circunferências concêntricas no ar. O sistema de rastreamento opera sobre essa sequência, na tentativa de acompanhar o movimento do alvo. A figura 4.11 mostra um quadro do vídeo de avaliação; são visíveis a elipse produzida pelo *camshift* e as posições das estimativas preditoras.

Para simular a presença de um ruído aditivo produzido por vibração da câmera, a cada quadro somou-se, à posição determinada pelo *camshift* para o alvo, um número inteiro aleatório de pixels, tanto na direção vertical quanto na direção horizontal. Esse número está compreendido no intervalo  $[-40, 40]$ . O erro de predição no quadro  $i$ , em pixels, é calculado como

$$E_i = \|s_i - \hat{s}_{i|i-1}\|, \quad (4.19)$$

onde  $s_i = [x_i \ y_i]^T$  é a posição corrente do alvo e  $\hat{s}_{i|i-1}$  é a estimativa preditora produzida para a mesma no quadro imediatamente anterior. O erro foi aferido para os 80 últimos quadros do vídeo, em uma série de 100 execuções distintas. Dessa



Figura 4.11: Sistema de acompanhamento operando sobre o vídeo utilizado para a avaliação de desempenho.

forma, foi obtida uma curva em que cada um dos 80 pontos representa a média do erro sobre 100 execuções, para o quadro correspondente.

O tempo de processamento de quadro compreende o intervalo entre a captação do quadro e o término do cálculo da posição do alvo e das previsões para o quadro subsequente, o que inclui a conversão da imagem para o espaço HSV, a execução da projeção reversa de histograma e a execução dos algoritmos *camshift* e dos preditores. De forma idêntica ao que foi feito para o registro do erro, o tempo foi medido para os 80 últimos quadros do vídeo, em uma série de 100 diferentes execuções. Foram avaliados os cenários em que nenhum preditor se encontra ativo e em que cada um deles está ativo individualmente.

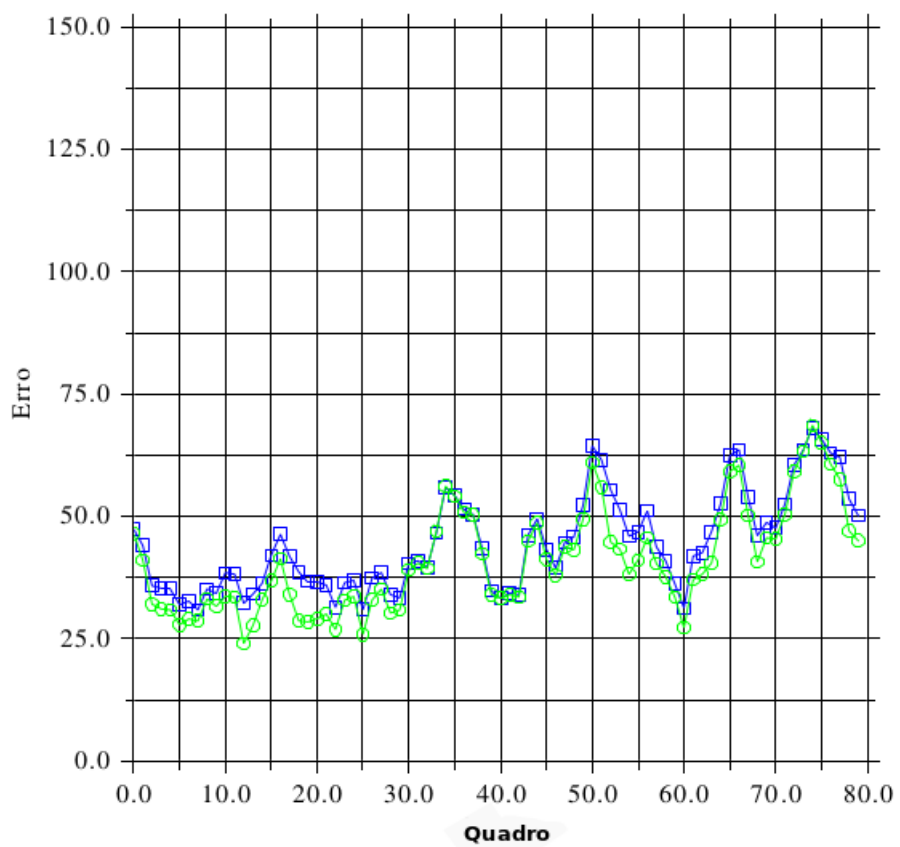
A próxima subseção apresenta os resultados obtidos no processo de avaliação de desempenho do sistema.



#### 4.5 Resultados e discussão

O gráfico da figura 4.12 mostra os erros médios de predição, em pixels, aferidos para os estimadores nominal e robusto, em cada um dos 80 últimos quadros do vídeo de teste. É possível observar ligeira vantagem do estimador robusto, que apresentou erro médio, sobre todos os quadros, de 40,92 pixels; para o estimador nominal, por sua vez, o erro médio foi de 44,39 pixels, cerca de 8,5 % maior. Embora a diferença de desempenho não seja expressiva, ela aponta o valor do maior nível de robustez do estimador, em presença de ruído de vibração da câmera. A superioridade do estimador robusto se faz mais notável em trechos onde o alvo se move com maior velocidade.

As figuras 4.13, 4.14 e 4.15 fornecem, respectivamente, os tempos de processamento de quadro medidos com nenhum dos preditores ativo, com apenas o preditor nominal ativo e com apenas o preditor robusto ativo. Como esperado, o tempo para a situação em que não há estimadores ativos é significativamente menor, com média de 1,47 ms. O preditor nominal produziu um tempo médio de 4,38 ms, e, o robusto, de 4,55 ms; ambos apresentam complexidade computacional muito semelhante, portanto. Os dois estimadores apresentaram desempenho bastante satisfatório, se mostrando adequados a aplicações que exigem processamento em tempo real; a robustez pode ser atingida sem imposições severas de tempo de processamento adicional sobre o caso nominal.



Media: 44.39

Media: 40.92

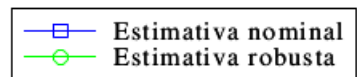
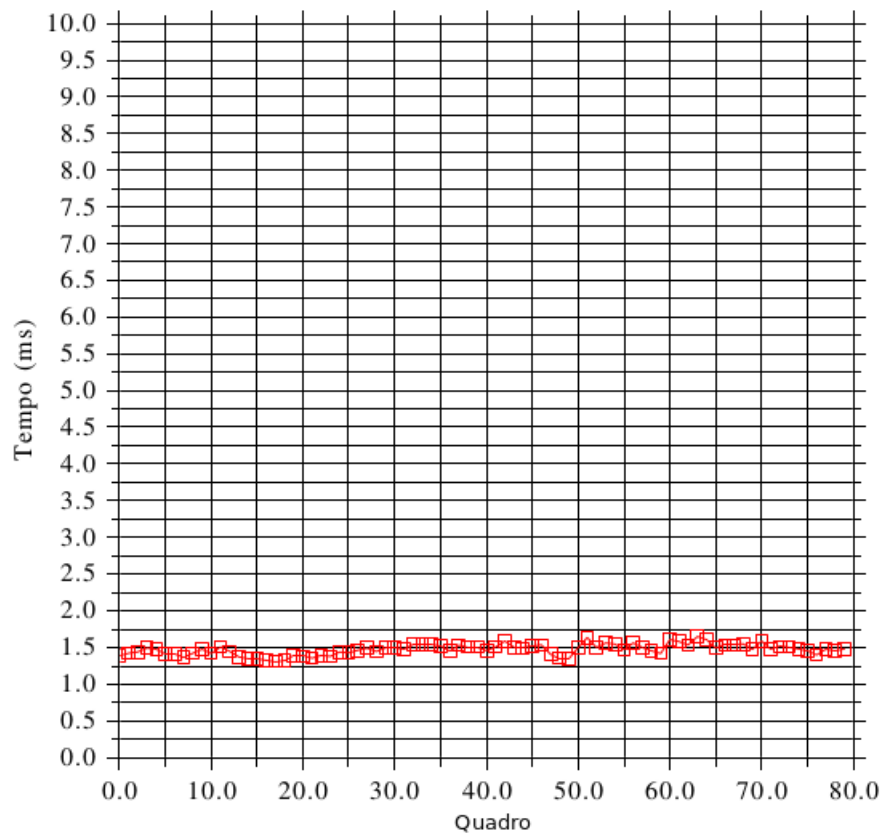
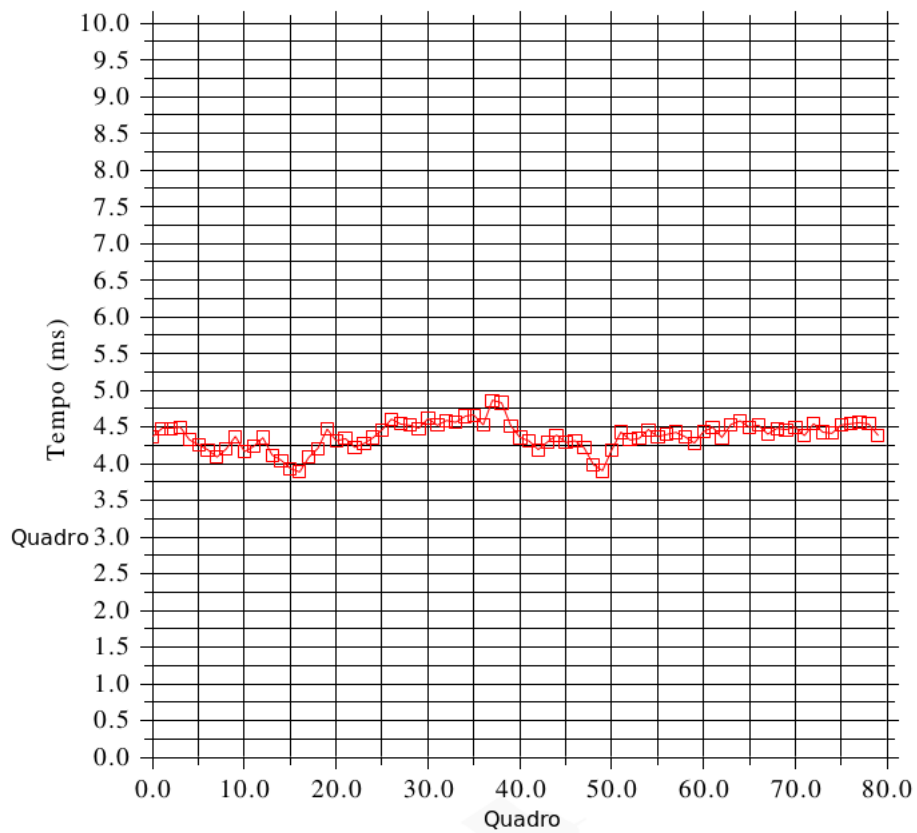


Figura 4.12: Erros de predição para os estimadores nominal e robusto, em pixels. Cada ponto da curva representa a média sobre uma série de 100 execuções da avaliação.



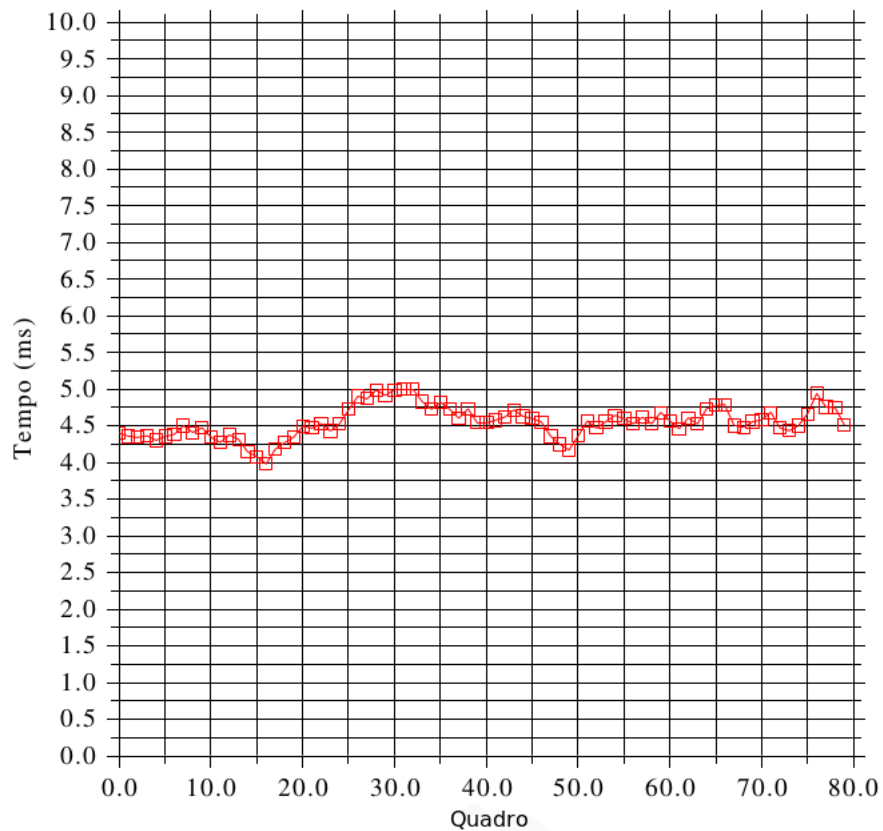
Media: 1.47

Figura 4.13: Tempo de processamento de quadro (ms) sem nenhum preditor ativo. Cada ponto da curva representa a média sobre uma série de 100 execuções da avaliação.



Media: 4.38

Figura 4.14: Tempo de processamento de quadro (ms) com somente o preditor nominal ativo. Cada ponto da curva representa a média sobre uma série de 100 execuções da avaliação.



Media: 4.55

Figura 4.15: Tempo de processamento de quadro (ms) com somente o preditor robusto ativo. Cada ponto da curva representa a média sobre uma série de 100 execuções da avaliação.

#### 4.6 Deficiências do sistema e dificuldades encontradas

Embora tenha apresentado desempenho satisfatório em diversos testes conduzidos, o sistema possui certas deficiências. A principal delas é a dificuldade que o algoritmo de segmentação encontra para identificar o alvo quando o mesmo se aproxima de outros objetos com cores semelhantes à sua. Uma situação como essa pode fazer com que o alvo seja confundido com o fundo da cena ou com objetos ao seu redor, como na figura 4.16. É possível minimizar o problema através de uma calibração das faixas de valores de S e de V consideradas, com

os seletores que se encontram no topo da janela. A imagem de distribuição de probabilidade de cor mostra as regiões da cena que são consideradas como parte do alvo, e portanto é útil observá-la durante o processo de calibração. Na figura 4.16, por exemplo, a distribuição revela que o objeto ao fundo de fato possui alta probabilidade de pertencer ao alvo. Após a calibração, o problema é solucionado, conforme mostra a figura 4.17. Isso porque o alvo possui uma cor mais escura, e, portanto, com menor valor de  $V$  em relação ao objeto ao fundo.

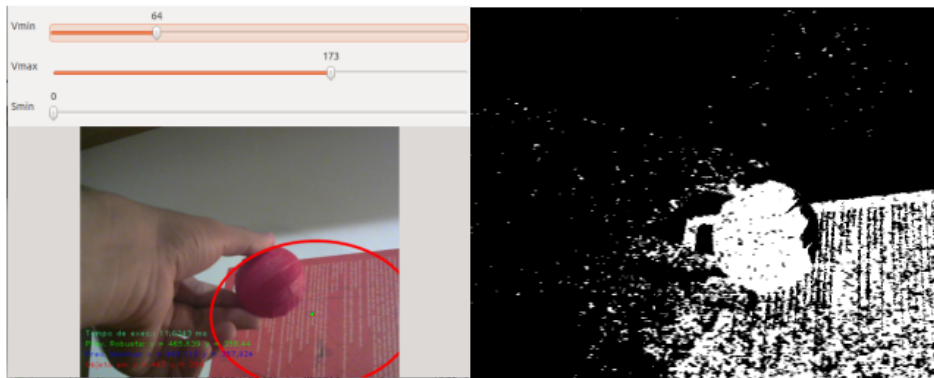


Figura 4.16: Situação em que o alvo (bola vermelha) se confunde com um objeto de cor semelhante ao fundo. À direita, a distribuição de probabilidade de cor do quadro.

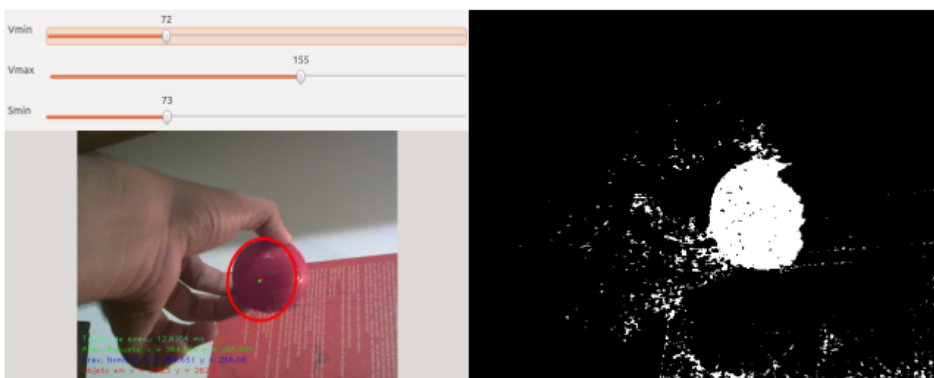


Figura 4.17: Após a calibração das faixas de  $S$  e  $V$  consideradas, o alvo não mais se confunde com o objeto ao fundo. À direita, a distribuição de probabilidade de cor do quadro.

Existe também certa dificuldade para que sejam determinados convenientemente certos parâmetros dos estimadores, como as matrizes  $Q$ ,  $R$ ,  $E_f$  e  $E_g$ . Em [15], foi utilizado um algoritmo evolutivo para a determinação dos valores. De fato, apesar de quase sempre permanecerem bastantes próximas à posição real do alvo, as estimativas frequentemente se encontram atrasadas em relação à mesma, como se não determinassem corretamente sua velocidade. Essa situação é visível na figura 4.18.



Figura 4.18: Predições (círculos azul e verde) atrasadas em relação à posição real do alvo (círculo vermelho).

#### 4.7 Desenvolvimentos futuros

Além de soluções para os inconvenientes citados na subseção anterior, outras adições e melhorias poderiam ser adequadas ao sistema. Por exemplo, a inclusão de algoritmos que busquem, de forma inteligente, os melhores valores para os parâmetros ajustáveis dos estimadores. Além disso, outras formulações de estimadores existem na literatura, e poderiam apresentar resultados mais satisfatórios. Também seria útil o acréscimo da capacidade de acompanhamento de múltiplos alvos simultaneamente, o que abriria margem para funções como detecção de colisão entre alvos. O reconhecimento e a classificação de alvos, como em [1], também seria uma adição interessante. Por fim, embora os preditores tenham sido avaliados e comparados quando empregados em conjunto com o algoritmo de rastreamento, o potencial dos mesmos como ferramenta adicional ao processo de acompanhamento não foi plenamente explorado.

Eles poderiam ser empregados, por exemplo, como um mecanismo para reduzir a janela de busca pelo alvo em cada quadro.



## 5 Conclusão

Neste trabalho, foram apresentados os principais aspectos de desenvolvimento e a avaliação de desempenho de um sistema de acompanhamento de um alvo móvel. Importantes conceitos relacionados ao processamento de imagens foram empregados, bem como resultados clássicos, e outros inovadores, da teoria de estimação linear.

Apesar de apresentar certas deficiências, o sistema se mostrou eficiente ao acompanhar o movimento do alvo selecionado pelo usuário na cena. Mesmo com os preditores ativos, o tempo de processamento de quadro não ultrapassa os 10 ms, de forma que não é severa a demanda do sistema por recursos computacionais. Os estimadores demonstraram ser satisfatoriamente bem-sucedidos na tarefa de prever a posição do alvo no quadro subsequente, sendo que a vantagem ficou por conta do filtro robusto, sobre o tradicional filtro de Kalman, por apresentar erros de predição ligeiramente menores em presença de ruído de vibração da câmera, e com tempo de processamento praticamente igual ao do outro.

Diversas melhorias e adições poderiam ser incorporadas ao sistema, sendo algumas delas de aplicação praticamente imediata. Ainda que demande ajustes na qualidade da segmentação e do rastreamento, e que um uso realmente efetivo dos preditores ainda esteja por ser feito, o sistema pode ser encarado como uma plataforma promissora para futuros desenvolvimentos.

## Referências

- [1] R. S. Patil A. J. Lipton, H. Fujiyoshi. *Moving target classification and tracking from real-time video*. Applications of Computer Vision, Princeton, 1998.
- [2] D. P. Bertsekas. *Dynamic programming and optimal control*, volume I. Athena Scientific, Belmont, Massachusetts, 1995.
- [3] G. R. Bradski. *Computer vision face tracking for use in a perceptual user interface*. Intel Technology Journal Q2, 1998.
- [4] L. Xia C. Hongxia, G. Qingchang. *Object tracking algorithm based on camshift algorithm combining with difference in frame*, volume 2. Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, 2007.
- [5] J. C. Teles Campos. *Filtragem robusta para sistemas singulares discretos no tempo*. Tese de doutorado, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2004.
- [6] H. Huosheng G. Dongbing. *Distributed minimax filter for tracking and flocking*. The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, 2010.
- [7] A. W. M. Smeulders H. T. Nguyen. *Fast occluded object tracking by a robust appearance filter*, volume 26. IEEE Transactions on Pattern Analyses and Machine Intelligence, 2004.
- [8] N. Papanikolopoulos H. Veeraraghavan, P. Schrater. *Robust target detection and tracking through integration of*

*motion, color, and geometry*, volume 103. Computer Vision and Image Understanding, Minneapolis, 2006.

- [9] Y. Xu M. Cohen J. Wang, B. Thiesson. *Image and video segmentation by anisotropic kernel mean shift*. Microsoft Research, Redmond, 2004.
- [10] L. D. Hostetler K. Fukunaga. *The estimation of the gradient of a density function, with applications in pattern-recognition*, volume IT-21. IEEE Transactions on Information Theory, 1975.
- [11] L. Jianxun L. Yongchen. *Robust adaptive kalman filtering for target tracking with unknown observation noise*. 2012 24th Chinese Control and Decision Conference (CCDC), 2012.
- [12] R. Canals N. Ould-Dris, A. Ganoun. *Improved object tracking with camshift algorithm*, volume 2. Acoustics, Speech and Signal Processing, 2006.
- [13] R. E. Woods R. C. Gonzales. *Digital image processing*. Pearson Prentice Hall, 2007.
- [14] A. H. Sayed S. M. Dominguez, T. Keaton. *Comparison of robust estimation and kalman filtering applied to fingertip tracking in human-machine interfaces*. Proc. Asilomar Conf. Signal, Systems & Computers, Pacific Grove, CA, 2001.
- [15] A. H. Sayed S. M. Dominguez, T. Keaton. *A robust finger tracking method for multimodal wearable computer interfacing*, volume 8. IEEE Transactions on Multimedia, 2006.
- [16] A. H. Sayed. *A framework for state-space estimation with uncertain models*. IEEE Transactions on Automatic Control, Universidade da Califórnia, 2001.

- [17] B. Hassibi T. Kailath, A. H. Sayed. *Linear estimation*. Prentice-Hall, Universidade da Califórnia, 2000.
- [18] A. H. Sayed T. Keaton, S. M. Dominguez. *Browsing the environment with the SNAP & TELL wearable computer system*, volume 9. Personal and Ubiquitous Computing J., 2005.

