Universidade Federal Fluminense
Escola de Engenharia
Mestrado em Engenharia de Telecomunicações

Wagner Marini de Oliveira

ESTABILIZAÇÃO DE VÍDEO DIGITAL GERADO A PARTIR DE CÂMERAS MÓVEIS UTILIZANDO *PARTICLE FILTER*

Niterói - RJ

2012

WAGNER MARINI DE OLIVEIRA

ESTABILIZAÇÃO DE VÍDEO DIGITAL GERADO A PARTIR DE CÂMERAS MÓVEIS UTILIZANDO *PARTICLE FILTER*

Dissertação apresentada ao Curso de Mestrado em Engenharia de Telecomunicações da Universidade Federal Fluminense - UFF, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Processamento de Sinais.

Orientador: Prof. ALEXANDRE SANTOS DE LA VEGA, D.Sc.

Niterói - RJ

2012

WAGNER MARINI DE OLIVEIRA

ESTABILIZAÇÃO DE VÍDEO DIGITAL GERADO A PARTIR DE CÂMERAS MÓVEIS UTILIZANDO *PARTICLE FILTER*

Dissertação apresentada ao Curso de Mestrado em Engenharia de Telecomunicações da Universidade Federal Fluminense - UFF, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Processamento de Sinais.

Aprovada em OUTUBRO de 2012.

BANCA EXAMINADORA

Prof. ALEXANDRE SANTOS DE LA VEGA, D.Sc. - Orientador
UFF

Prof. MURILO BRESCIANI DE CARVALHO, D.Sc.
UFF

Prof. TADEU FERREIRA, D.Sc.
UFF

Prof. GELSON VIEIRA MENDONÇA, Ph.D.
UFRJ

Niterói - RJ
2012

Dedico este trabalho ao meu filho Victor Hugo pelo incentivo e longas horas de discussão sobre este conteúdo. Seu interesse me motiva a aprender e ensinar sempre e sempre.

Agradecimentos

Ao meu orientador Alexandre Santos de la Vega pelo seu suporte, boa vontade e incentivo sempre valioso neste trabalho.

Ao Amir Said que me abriu um canal de diálogo fundamental para a definição deste trabalho.

Ao amigo André Resende pelo incentivo a me candidatar nesse programa de Mestrado.

Ao meu filho e meu assistente de pesquisas Victor Hugo pela sua participação em diversos temas desta dissertação.

Aos meus pais por sempre acreditarem em mim e torcerem pelo meu sucesso.

Aos professores desse programa de Mestrado pela sua dedicação, trabalho e esforço na profissão de docente, e que, de alguma forma, direta ou indiretamente, me ajudaram no desenvolvimento deste trabalho.

A Deus que me presenteou com a vontade de querer humildemente entender sua criação pela ótica da Engenharia.

Lista de Figuras

Lista de Tabelas

Sumário

Resumo

Com a popularidade cada vez maior de câmeras embarcadas em dispositivos móveis, torna-se necessário avaliar técnicas que resolvam alguns problemas inerentes a esse tipo de geração de vídeo digital em cores. Este trabalho propõe uma técnica robusta para estabilização de vídeo, utilizando a metodologia *Particle Filter*, que tem origem na filtragem Bayesiana e que é empregado em problemas de acompanhamento não linear e, possivelmente, não-Gaussiano. Neste tipo de filtragem, utilizando um mecanismo Monte Carlo Sequencial, são geradas amostras aleatórias do vetor de estados, seguindo uma distribuição de probabilidades conhecida. É atribuído para cada amostra um peso, ou importância, permitindo simular recursivamente a distribuição da densidade *a posteriori* não linear. Com o foco em aplicações para videoconferência, esta técnica acompanha, quadro a quadro, a face selecionada do interlocutor, modelada como uma elipse parametrizada, mesmo em um cenário de grande variação da sua localização devido a algum distúrbio indesejado, provocado por uma instabilidade na geração do vídeo. Utilizando um algoritmo de baixo custo, cada quadro é dividido em diferentes blocos não sobrepostos, onde os histogramas de cor de suas bordas são comparados a um modelo previamente selecionado da face a ser acompanhada, possibilitando a detecção da sua nova localização e tamanho de forma rápida. A posição estimada da face define uma região de importância que determina qual a distribuição o algoritmo *Particle Filter* aplica para as amostras aleatórias necessárias para estimar o novo vetor de estado da face do interlocutor. Este método foi batizado pelo nome Blocos Particionados (BP) e, em conjunto com as técnicas de Módulo Gradiente e Módulo de Cor, integram as técnicas propostas neste trabalho para acompanhar a face quadro a quadro. Com a localização precisa da face acompanhada, aplica-se uma transformação linear no quadro corrente, compensando a variação indesejada que ocasiona a instabilidade do vídeo.

Palavras-chave: Estabilização de vídeo, Acompanhamento, Visão Computacional, *Particle Filter*, Monte Carlo Sequencial.

Abstract

With the increased popularity of embedded cameras in mobile devices, new approaches and techniques on how to generate fairly good quality video became necessary to be investigated to solve some inner problems that users has been experienced when recording color movies without a stabilized platform. This work proposes a robust and real-time technique to stabilize video based on Particle Filter framework. This methodology has its inspiration from Bayesian filtering which has in tracking and stochastic estimation some of the core applications due to their nonlinear and non-Gaussian behaviours. This kind of filtering generates random samples with a known probability distribution following Sequential Monte Carlo algorithms to simulate the a posteriori probability function and then estimates the new state vector of the dynamic system. Teleconferencing applications are the principal goal to be handled by this work. Processing an input video with some undesirable jitter throw this filter generates an stabilized output as the expected video result. By taking advantage from an importance sampling based on motion cues, this work gets a quick processing. The motion cues use an approach based on color histograms generated from a very small set of pixels taken from each incoming frame that finds the current face position defining a displacement vector. Particle filter uses this displacement vector to refine the current face state and eventually apply the desired statilization on the video.

Keywords: Video Stabilization, Tracking, Computer Vision, Particle Filter, Sequential Monte Carlo.

Capítulo 1

Introdução

1.1 Motivação

O desenvolvimento tecnológico no setor de Telecomunicações tem realizado vigoroso progresso na comunicação pessoal nos últimos cinco anos, o que vem impactando de forma significativa o modo de se estabelecer contato entre pessoas. O uso de diversas mídias, não restritas apenas à voz, numa mesma chamada telefônica está contribuindo para uma aproximação cada vez mais real dos interlocutores. É possível que indivíduos em diferentes regiões geográficas compartilhem o mesmo ambiente virtual e tenham a capacidade de se comunicar de forma interativa, como se estivessem no mesmo ambiente físico, permitindo até a troca de documentos digitais e a comunicação visual.

Alguns produtos já disponíveis no mercado, que possuem avançados recursos de teleconferência com multimídia, têm ainda um foco restrito a ambientes corporativos ou especialistas, não somente devido à necessidade de uma exigente e complexa infraestrutura de telecomunicações, mas também por serem carentes de soluções adaptadas ao usuário que não usufrui dos recursos exigidos para tal, como por exemplo o usuário móvel.

No que se refere aos avançados recursos de infraestrutura em telecomunicações, estão dois pontos de real importância: (i) a exigência de uma capacidade de banda de transmissão que seja confiável e que permita o fluxo de grande capacidade de dados em tempo real, sem a percepção por parte do usuário de atrasos que ocasionam o total colapso no sentimento de reciprocidade na comunicação estabelecida e (ii) a codificação e a decodificação do áudio e do vídeo realizadas de forma robusta e que permita, ao mesmo tempo, a utilização de uma menor capacidade de recursos de transmissão possível, mesmo

que exija uma grande capacidade de processamento.

Nos ambientes corporativos e especialistas, encontramos todos esses recursos descritos acima no estado da arte. Podemos citar, como exemplo, uma intervenção cirúrgica assistida, que é um ambiente extremamente especializado por onde podem ser executadas as cirurgias remotamente. Outro exemplo é a cooperação em projetos de desenvolvimento e pesquisa multinacionais.

O tema para o qual esse trabalho busca contribuir está focado na investigação do uso de tecnologia móvel para aplicações em videoconferência. Assim, é necessário analisar os requisitos que permitam o uso desta tecnologia, propondo uma solução de estabilização em tempo real do vídeo gerado por uma câmera móvel, sendo possível utilizá-la em aplicações que exigem interatividade. Os requisitos qualitativos que este trabalho busca atender são:

- disponibilizar vídeo em alta definição (HD, 1080p) ou menor;
- mesmo em deslocamento, disponibilizar, sempre que possível, vídeo estabilizado, ou seja, sem a trepidação natural do movimento da mão ou do veículo em que a câmera se encontra;
- atender aos requisitos de capacidade de banda de transmissão em tempo real e robustez;
- a região de interesse deve conter em seu centro a face do usuário, exigindo assim que haja um acompanhamento (*tracking*) por parte da câmera. Se houver mais de um interlocutor compartilhando a mesma câmera, é necessário o acompanhamento individual de cada região de interesse de forma independente.

Ao disponibilizar-se vídeo em alta definição, permite-se atender a evolução natural da tecnologia, que, eventualmente, irá oferecer este tipo de recurso até em dispositivos móveis. Este requisito, associado ao requisito de tempo real, que será abordado a seguir, exige que se tenha em mente elaborar algoritmos rápidos para os diversos problemas de computação visual.

O vídeo estabilizado é necessário e compõe a parte mais importante da pesquisa. Com ele se consegue oferecer aos interlocutores uma imagem confortável do vídeo gerado pelo interlocutor móvel. Isso permite que a teleconferência seja conduzida de forma

natural pelas partes, sem a necessidade, por parte do interlocutor móvel, de estabilizar manualmente sua imagem.

Os requisitos de tempo real e robustez são necessários devido à natureza da teleconferência. Retardos no processamento da imagem dos interlocutores geram atraso na comunicação e transmissão do vídeo e inviabilizam a teleconferência.

A região de interesse com foco na face do interlocutor móvel torna ainda mais natural o diálogo entre os interlocutores. Um acompanhamento da face é um recurso vital para que o interlocutor móvel tenha facilidade de se expressar e ser entendido pelos seus pares.

Assim, em resumo, o principal problema a ser identificado e solucionado nesta monografia é a estabilização de imagem, além do reconhecimento e acompanhamento de face do interlocutor móvel, de modo que seja natural a sua interação num sistema de teleconferência com os outros interlocutores.

1.2 Cenários previstos

Para definir quais as técnicas utilizadas na construção do algoritmo de estabilização de vídeo e de acompanhamento de objetos (3D *tracking*), algumas restrições ao uso devem ser aplicadas em relação aos seguintes itens:

- complexidade de processamento do vídeo;
- resolução máxima do vídeo;
- tempo de processamento de cada quadro do vídeo;
- objeto de interesse.

Tais restrições são definidas a seguir.

Complexidade de processamento de vídeo

O cenário de menor complexidade ocorre quando a câmera estiver fixa, em ambiente fechado, com baixa variação do fundo da cena), com iluminação constante e somente o

objeto de interesse, a face do interlocutor, move-se. Neste cenário, é possível inibir o algoritmo de processamento global da cena e, consequentemente, o algoritmo de estabilização da imagem responsável pela detecção do movimento de *shaking*.

Quanto ao cenário de maior complexidade, o mesmo ocorre quando a câmera não estiver fixa, como, por exemplo, na mão do interlocutor, num ambiente aberto (*outdoor*), com grande variação do fundo da cena e iluminação variando conforme a translação do plano. Neste cenário o algoritmo deverá reconhecer e acompanhar o objeto de interesse, além de calcular de forma robusta o movimento global da cena que corresponde ao movimento de trepidação da câmera e, consequentemente, gerando um vídeo de saída estabilizado.

Resolução do vídeo e tempo de processamento de cada quadro

O algoritmo deve atender, no máximo, à resolução de vídeo de 1080p a 60 quadros/s.

Objetos de interesse

O principal objeto de interesse para este trabalho é a face do interlocutor. É possível que ocorra um cenário onde diversos interlocutores compartilham a mesma câmera de vídeo. Este cenário será atendido quando não houver a necessidade de processamento adicional para o cálculo da trepidação do vídeo, ou seja, quando a câmera estiver fixa e somente a cena (interlocutores e fundo) se movimentam. A cada face reconhecida existe um processamento distinto que é tratado separadamente.

1.3 Revisão bibliográfica

A utilização de metodologias de filtragem digital de sistemas dinâmicos em tempo discreto baseado em filtragem Bayesiana, como a técnica *Particle Filter*, foi recentemente migrada da área da Estatística para emprego em soluções de problemas em Engenharia, entre elas, aplicações envolvendo o processamento digital de sinais.

O *Particle Filter* tem sua classificação baseada nas diferentes estratégias sugeridas para a amostragem da densidade proposta ou função importância, $q(\cdot)$. A Tabela 1.3 apresenta os nomes e as referências bibliográficas das técnicas mais conhecidas.

Tabela 1.1: Classificação das técnicas *Particle Filter*.

Sequential Importance Sampling (**SIS**)	A técnica base para as demais técnicas baseadas em *Particle Filter*. Representa a densidade *a posteriori* $p(\mathbf{x}_k\|\mathbf{z}_{1:k})$ a partir de um conjunto de amostras de estados selecionadas aleatoriamente e atribui a cada amostra um peso que identifica a sua importância nesta distribuição.	[IB98], [AMGC02]
Sampling Importance Resampling (**SIR**)	Utiliza a densidade *a priori* $p(\mathbf{x}_k\|\mathbf{x}^i_{k-1})$ como densidade proposta.	[AMGC02]
Auxiliary Sampling Importance Resampling (**ASIR**)	Utiliza a densidade *a priori* $p(\mathbf{x}_k\|\mathbf{x}^i_{k-1})$ e a densidade de verossimilhança $p(\mathbf{z}_k\|\mathbf{x}^i_k)$ como densidade proposta.	[PS99]
Regularized Particle Filter (**RPF**)	Semelhante ao SIR. Porém, para evitar o problema de degeneração que nele ocorre, realiza o estágio de reamostragem numa distribuição *a posteriori* $p(\mathbf{x}_k\|\mathbf{z}_{1:k})$ contínua ao invés de discreta.	[DdFG01]

Em recentes pesquisas do tema *Particle Filter* na área de acompanhamento de objetos (*object tracking*) em Visão Computacional, o trabalho publicado por Michael Isard e Andrew Blake, *Condensation*, [IB98], é considerado pela comunidade científica como o precursor desta linha de abordagem. Nele, foi apresentado a técnica *Particle Filter* empregando a densidade de probabilidade *a priori* para a definição da densidade proposta (*proposal density*), obtendo resultados bastante satisfatórios. Seu algoritmo implementa a amostragem da densidade proposta a partir de duas distribuições previamente conhecidas no instante de tempo k: (i) a estimativa do estado de $\mathbf{x}_k$ a partir do conjunto de observações do sistema até o instante de tempo $k-1$, $\mathbf{z}_{1:k-1}$, conhecida por $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$, que é obtida do conhecimento da distribuição *a posteriori* no instante $k-1$, dado por $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$; (ii) a distribuição *a priori* da transição marginal de estados entre os instantes de tempo $k-1$ e k, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. Observando ambas as distribuições, não há referência à utilização do vetor de observação $\mathbf{z}_k$ na definição do conjunto partículas/peso

(*particles/weight*) que defina a distribuição proposta $q(\cdot)$. Uma possível otimização deste método ocorre quando a informação sobre $\mathbf{z}_k$ é relevante ao processamento do filtro. Incluindo esta distribuição no cálculo da estimativa da distribuição *a posteriori* permite maior robustez e acurácia na implementação.

[BS04] utiliza a técnica SIS para o acompanhamento de cabeça numa sequência de vídeo digital. Sua proposta considera o movimento da cabeça inter-quadros na geração da *Importance Function* através de um algoritmo de segmentação de cada quadro do vídeo em blocos que são classificados como blocos sementes, incertos ou *background*. Utilizando um algoritmo de busca e comparação *block matching*, o deslocamento inter-quadros dos blocos sementes e incertos são calculados e, então, são identificadas as regiões no quadro corrente onde se encontra a cabeça acompanhada. Este trabalho modela a cabeça a ser acompanhada por um vetor de estados de quatro dimensões que foi também utilizado neste trabalho. Esta dissertação recebeu grande influência desse trabalho quanto a técnica de construção da função de importância, a modelagem da face como uma elipse parametrizada e, por fim, na implementação do algoritmo baseado na metodologia *Particle Filter*.

[Yan09] apresenta uma técnica de estabilização robusta de vídeo digital utilizando *Particle Filter* para estimação e acompanhamento do movimento global da câmera sem estabilização. O trabalho propõe um modelo de estados baseado em um conjunto de seis parâmetros que definem as transformações espaciais de coordenadas aplicadas sucessivamente aos quadros. As transformações previstas em seu modelo são: rotação, translação e escalonamento. Uma vez realizada a estimação, aplica-se uma operação inversa, que resulta na estabilização prevista do conteúdo de vídeo. Para identificar a *Importance Function*, utiliza-se a técnica *Scale-Invariant Feature Transform* (SIFT). O tema mais importante de [Yan09] para o desenvolvimento desta dissertação foi a explanação consistente e suscinta sobre os conceitos teóricos que permeiam a técnica *Particle Filter* aplicados ao acompanhamento e à estabilização de vídeo digital.

No trabalho [PYLJ09] é proposto um algoritmo de estabilização de vídeo tentando eliminar o efeito indesejável típico de câmeras móveis. Sua abordagem apesar de bastante simples e compreensível utiliza um algoritmo baseado em *Optical Flow* com referência a abordagem clássica do método Kandade-Lucas-Tomasi (KLT). É mostrado que a estimação do movimento indesejável da câmera pode ser extraído do movimento global da imagem inter-quadros, tema utilizado nesta dissertação.

Basicamente, os trabalhos relatados acima contribuíram de alguma forma para a construção do método proposto nesta dissertação. Há ainda alguns trabalhos acessórios que foram utilizados para resolver alguns pontos importantes da implementação do algoritmo. Entre eles estão o [SB91], que apresenta a modelagem de cor e seus histogramas úteis para a detecção de objetos numa cena; o [FPF99], que implementa o algoritmo *least-squares* para encontrar uma elipse a partir de pontos esparsos num plano e os tutoriais apresentados por [AMGC02] e também [DJ09], fundamentais para a construção dos fundamentos necessários para a compreensão das metodologias *Sequential Importance Sampling* e *Particle Filter*.

1.4 Organização da dissertação

Tendo como foco principal a pesquisa voltada à área de Processamento de Sinais em Sistemas de Multimídia, esta monografia busca propor uma solução ao problema de estabilização de vídeo digital empregando a técnica de filtragem digital por *Particle Filter*.

No Capítulo 2 é apresentado uma revisão teórica dos fundamentos utilizados na metodologia de filtragem digital sub-ótima por *Particle Filter*, concluindo o capítulo com o algoritmo detalhado de sua implementação.

O Capítulo 3 é dedicado à apresentação da modelagem da face (objeto de interesse da filtragem) em uma sequência de vídeo digital como objeto de acompanhamento do sistema. Será discutido como a face é modelada por uma elipse parametrizada, identificando cada dimensão do vetor de estados.

O Capítulo 4 apresenta a definição da dinâmica dos elementos da cena, dando uma importância inicial à descrição das equações de estados do sistema dinâmico, discreto no tempo e, em seguida, à aplicação dos conceitos teóricos apresentados no Capítulo 2 no modelo proposto, sob a ótica da técnica de *Particle Filter*.

No Capítulo 5 é apresentada a implementação da estabilização do vídeo a partir do deslocamento de uma janela flutuante, que se desloca na mesma amplitude estimada para o deslocamento da face, mantendo assim o vídeo estabilizado e proporcionando uma melhoria na relação sinal-ruído, em comparação ao vídeo original de entrada.

No Capítulo 6 são apresentados os resultados experimentais para cada uma das técnicas descritas e aplicadas nesta dissertação. São apresentados diversos cenários, os

resultados obtidos e a comparação dos mesmos com os critérios adotados pela comunidade científica, permitindo assim a apresentação de resultados comparativos com outros estudos publicados .

O Capítulo 7 apresenta as conclusões observadas na aplicação desta metodologia para resolver o problema de estabilização do vídeo digital e a sugestão de trabalhos futuros, dando continuidade a este estudo.

Informações adicionais são apresentadas em apêndices. No Apêndice A a nomenclatura utilizada em todo o texto. No Apêndice B, a descrição teórica do Processo Gaussiano, amplamente discutido na exposição dos conceitos referentes à filtragem digital Bayesiana. No Apêndice C, a demonstração algébrica de equações apresentadas neste trabalho. No Apêndice D, as imagens de teste utilizadas para a apresentação dos resultados experimentais.

Capítulo 2

Fundamentos do *Particle Filter*

2.1 Introdução

Particle Filter é uma excelente ferramenta para modelar sistemas dinâmicos não lineares e vem ultimamente recebendo bastante atenção na solução de problemas de engenharia e de estatística que envolvem acompanhamento e estimação através do uso de um modelo de espaço de estados, markoviano e em tempo discreto.

A partir da metodologia Monte Carlo sequencial, o *Particle Filter* procura recursivamente aproximar a filtragem Bayesiana através da construção de um conjunto de pontos de massas, conhecidos como *particles*, usando uma distribuição de probabilidade conhecida e realizável, chamada de *importance distribution* ou *proposal distribution*. A *importance distribution* representa uma densidade de probabilidade que é proporcional à densidade de probabilidade *a posteriori* do vetor de estados que modela o objeto de acompanhamento desejado, que, por sua natureza intrínseca, poderá ser não-linear e ainda não-gaussiana.

Neste capítulo é apresentado uma introdução teórica sobre a técnica *Particle Filter*, onde cada uma das seções apresenta os seguintes assuntos:

1. uma representação no espaço de estados, visando um modelo que represente um sistema dinâmico, não-linear, no tempo discreto;

2. uma introdução à amostragem Bayesiana, fundamental para a construção de um modelo probabilístico utilizado pela técnica;

3. uma introdução à filtragem Monte Carlo sequencial, que possibilita a estimação das partículas (*particles*) que formam o modelo estatístico do problema de estimação;

4. uma introdução ao arcabouço teórico necessário para descrever matematicamente a técnica *Particle Filter*.

2.2 Espaços de estados

O problema de acompanhamento da face proposto nesta dissertação pode ser analizado como um problema de estimação não-linear. Para se chegar a esta conclusão é apresentado, neste tópico, uma descrição da estrutura de modelagem dos dados utilizada.

Um sistema dinâmico discreto em que se deseja aplicar uma filtragem digital requer um modelo matemático que o descreva em função do tempo. Encontrar um modelo adequado e compacto permite representar um sistema com **q**-dimensões em um espaço de **p**-dimensões, $\mathbf{p} < \mathbf{q}$, diminuindo assim a complexidade da solução e possivelmente o esforço computacional que a solução proposta oferece.

Um modelo possui necessariamente dois componentes principais [MK04]:

1. uma equação de restrição ou de estado do sistema, e,

2. uma equação de observação ou de medida.

A equação de restrição descreve o conhecimento *a priori* do sistema. Ela representa como o sistema se comporta, enquanto que a equação de observação descreve como os dados são obtidos, muitas vezes através de uma medição imprecisa, que é representado no modelo como ruído aditivo ao sistema.

A restrição pode ser definida pelo conjunto de expressões algébricas

$$\psi_k(x_1, \ldots, x_m; \theta_1, \ldots, \theta_p) = \mathbf{0} \quad , \quad k = 1, \ldots, K, \tag{2.1}$$

onde $\mathbf{x} = \{x_1, \ldots \ x_m \,|\, \mathbf{x} \in \Re^m\}$ representa as funções bases, ou portadoras, do modelo proposto.

São normalmente funções $\varphi(\centerdot)$, monomiais e possivelmente não lineares, que identificam a dependência entre as variáveis de observação, representada pelo vetor $\mathbf{y} = \{y_1, \ldots, y_q \,|\, \mathbf{y} \in \Re^q\}$ com as variáveis de estado do sistema, tais como

$$x_j = \varphi(y_1, \ldots, y_q) = \varphi(\mathbf{y}) \quad , \quad j = 1, \ldots, m. \tag{2.2}$$

Já o conjunto $\boldsymbol{\theta} = \{\theta_0, \ldots, \theta_p \,|\, \boldsymbol{\theta} \in \Re^p\}$ representa os parâmetros que se deseja estimar no modelo.

A Equação (2.1) é uma equação de restrição bastante genérica para ser aplicada na definição do problema de estimação proposto aqui. Assim, definindo K = 1 e a dimensão das funções bases igual à dimensão dos parâmetros, $\mathbf{m} = \mathbf{p}$, teremos a relação escalar e linear entre as funções bases $\mathbf{x}$ e os parâmetros $\boldsymbol{\theta}$ dada por

$$\alpha + \mathbf{x}^T\boldsymbol{\theta} = 0 \tag{2.3}$$

e

$$\mathbf{x}^T = [\varphi_1(\mathbf{y}) \cdots \varphi_p(\mathbf{y})] \quad , \tag{2.4}$$

onde $\alpha = \theta_0$ representa o termo constante da equação linear.

Um exemplo de aplicação das Equações (2.3) e (2.4) ocorre quando se estima os parâmetros de uma elipse a partir de um conjunto de pontos, que representam os dados observados da elipse, considerando a existência de ruído na medida. Este exemplo é relevante nesta dissertação e será detalhado mais tarde no Capítulo 3, quando o estudo da modelagem da face como o objeto de acompanhamento será apresentado.

A equação de restrição que define a elipse pode ser escrita [Pet66] como

$$(\mathbf{y} - \mathbf{y}_c)^T\mathbf{Q}(\mathbf{y} - \mathbf{y}_c) - 1 = 0 \quad , \tag{2.5}$$

onde $\mathbf{y} = [y_1\ y_2]^T$ corresponde as coordenadas de cada ponto da elipse, $\mathbf{y}_c$ corresponde à coordenada do centro geométrico da elipse e a matriz real e simétrica

$$\mathbf{Q} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \quad ,$$

representa o termo quadrático da equação implícita da elipse que é apresentado na Equação (3.2).

A Equação (2.5) pode ser reescrita como

$$\mathbf{y}^T\mathbf{Q}(\mathbf{y} - \mathbf{y}_c) - \mathbf{y}_c^T\mathbf{Q}(\mathbf{y} - \mathbf{y}_c) - 1 = \mathbf{y}^T\mathbf{Q}\mathbf{y} - \mathbf{y}^T\mathbf{Q}\mathbf{y}_c - \mathbf{y}_c^T\mathbf{Q}\mathbf{y} + \mathbf{y}_c^T\mathbf{Q}\mathbf{y}_c - 1 = 0 \quad . \tag{2.6}$$

Uma vez que $\mathbf{y}^T\mathbf{Q}\mathbf{y}_c$ resulta em uma matriz 1×1, tem-se que

$$(\mathbf{y}^T\mathbf{Q}\mathbf{y}_c)^T = \mathbf{y}_c^T\mathbf{Q}^T\mathbf{y} \quad . \tag{2.7}$$

Dado que $\mathbf{Q} = \mathbf{Q}^T$, pode-se escrever que

$$-\mathbf{y}^T\mathbf{Q}\mathbf{y}_c - \mathbf{y}_c^T\mathbf{Q}\mathbf{y} = -2\mathbf{y}_c^T\mathbf{Q}\mathbf{y} \quad . \tag{2.8}$$

Portanto, a Equação 2.6 pode assumir a forma

$$\mathbf{y}^T\mathbf{Q}\mathbf{y} - 2\mathbf{y}_c^T\mathbf{Q}\mathbf{y} + \mathbf{y}_c^T\mathbf{Q}\mathbf{y}_c - 1 = 0 \quad . \tag{2.9}$$

A partir do resultado obtido na Equação (2.9), encontramos a equação de restrição tal como é apresentado em (2.3) e (2.4).

Do termo $\mathbf{y}^T\mathbf{Q}\mathbf{y}$, temos

$$\begin{aligned}
\mathbf{y}^T\mathbf{Q}\mathbf{y} &= \begin{bmatrix} y_1 & y_2 \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\
&= \begin{bmatrix} y_1Q_{11} + y_2Q_{21} & y_1Q_{12} + y_2Q_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\
&= y_1^2Q_{11} + y_1y_2Q_{21} + y_1y_2Q_{12} + y_2^2Q_{22} \quad .
\end{aligned} \tag{2.10}$$

Enquanto o termo apresentado em (2.8) define os parâmetros de primeira ordem da equação de restrição, o termo apresentado em (2.10) define os parâmetros de segunda ordem que são não lineares. Tomando a Equação 3.2 como base, podemos reescrever seus termos de segunda ordem como

$$\mathbf{y}^T\mathbf{Q}\mathbf{y} = \mathbf{y}^T \begin{bmatrix} a & \frac{b}{2} \\ \frac{b}{2} & c \end{bmatrix} \mathbf{y},$$

e assim encontrar a matriz simétrica Q igualando os termos $Q_{12} = Q_{21}$. Dessa forma as Equações (2.3) e (2.4) podem ser representadas por

$$\alpha = \mathbf{y}_c^T\mathbf{Q}\mathbf{y}_c - 1 \tag{2.11a}$$

$$\boldsymbol{\theta}^T = \begin{bmatrix} -2\mathbf{y}_c^T\mathbf{Q} & Q_{11} & 2Q_{12} & Q_{22} \end{bmatrix} \tag{2.11b}$$

$$\mathbf{x}^T = \begin{bmatrix} y_1 & y_2 & y_1^2 & y_1y_2 & y_2^2 \end{bmatrix} \quad . \tag{2.11c}$$

Quanto à equação de observação definida para o sistema, partindo de um modelo em que a leitura dos dados identificado pelo vetor $\mathbf{y}$ é ainda corrompida por um ruído aditivo independente $\delta\mathbf{y}_i$, pode-se chegar ao vetor $\mathbf{z}$ de observação dado por

$$\mathbf{z} = \mathbf{y} + \delta\mathbf{y}_i \quad . \tag{2.12}$$

Agora, é necessário apresentar um modelo dinâmico discreto para o sistema que permita uma solução iterativa da estimação dos seus parâmetros. Dessa forma, é possível, a partir das Equações de estado (2.3) e (2.4) e da Equação de observação (2.12), propor uma solução que estime recursivamente a densidade de probabilidade *a posteriori* $p(\mathbf{x}|\mathbf{z})$, como será apresentado mais tarde, na Seção 2.3, quando uma representação probabilística do modelo de espaço de estados, a partir das função de distribuição de probabilidades, é descrita.

Como apresentado por [AMGC02], dada a sequência de estados no tempo discreto representado por $\mathcal{X} = \mathbf{x}_k, k \in \mathbb{N}$, a equação de estado do sistema dinâmico pode ser descrita como

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) \quad , \tag{2.13}$$

onde, $\mathbf{f}_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_x}$ é uma função possivelmente não linear que modela o conhecimento *a priori* do sistema, ou seja, é a função que representa a transição do estado anterior (k-1) para o estado atual (k). A função $\mathbf{f}(\centerdot)$ ainda tem como parâmetro a sequência $\mathbf{v}_{k-1}$, $k \in \mathbb{N}$ que representa o processo de ruído independente e identicamente distribuído (i.i.d.). Os valores escalares n_x e n_v representam as dimensões no espaço Euclideano dos vetores de estado e de ruído, respectivamente.

A relação que o vetor de estado $\mathbf{x}_k$ possui com o vetor de observação $\mathbf{z}_k$, no tempo discreto, pode ser representada pela seguinte equação de observação:

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \eta_k) \quad , \tag{2.14}$$

onde, $\mathbf{h}_k : \Re^{n_x} \times \Re^{n_\eta} \rightarrow \Re^{n_z}$ é uma função possivelmente não linear. Esta função inclui ainda outro parâmetro de entrada, representado pelo vetor de ruído i.i.d. η_k, indicando que os valores amostrados da sequência $\mathbf{z}_k$ são corrompidos por uma parcela de ruído que pode ser considerado, para fins práticos, o erro de medida. Assim como na Equação

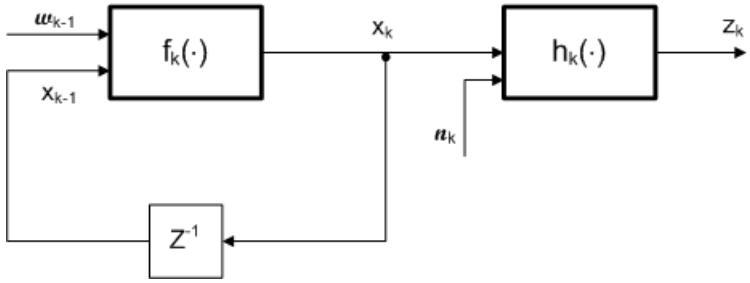

Figura 2.1: Modelo em espaço de estados de um sistema discreto no tempo, possivelmente não-linear.

(2.13), n_x e n_η, na Equação (2.14), representam as dimensões no espaço Euclidiano dos vetores de observação e de ruído, respectivamente.

A Figura 2.1 representa modelo do sistema dinâmico não linear, discreto no tempo, representado pelas Equações (2.13) e (2.14), utilizado na formulação do problema de acompanhamento e de estimação proposto nesta dissertação.

2.3 Acompanhamento bayesiano

Seguindo a modelagem por espaço de estados apresentado na Seção 2.2, a observação de uma sequência de amostras obtidas numa ótica temporal descreve como um sistema dinâmico comporta-se em função do tempo discreto k. A idéia por detrás desta modelagem no domínio do tempo discreto, $k \in \mathbb{N}$, permite construir sequencialmente um algoritmo de estimação dos estados do sistema baseado na sua distribuição conjunta de probabilidades envolvendo tanto o conjunto de todas as amostras da sequência de observação, $\{\mathbf{z}_{1:k}; k = 1, 2, \dots\}$, como também a sequência de estados do processo dinâmico, $\{\mathbf{x}_{0:k-1}; k = 1, 2, \dots\}$. A Figura 2.2 apresenta graficamente a relação entre os vetores de estado do sistema e de observação, com suas respectivas probabilidades conjunta num espaço de tempo discreto. A implementação de algoritmos sequenciais é de notória importância quando se deseja empregá-los em problemas de filtragem digital em tempo-real. A essa técnica dá-se o nome de Estimação Bayesiana Sequencial (*Sequential Bayesian Estimation*) [Can09].

Adotando uma abordagem Bayesiana clássica para o problema de estimação em processamento digital de sinais, procura-se encontrar uma solução ótima para o cálculo da distribuição conjunta de probabilidade *a posteriori* ($p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k});\ k = 0, 1, \dots$). Recorrendo

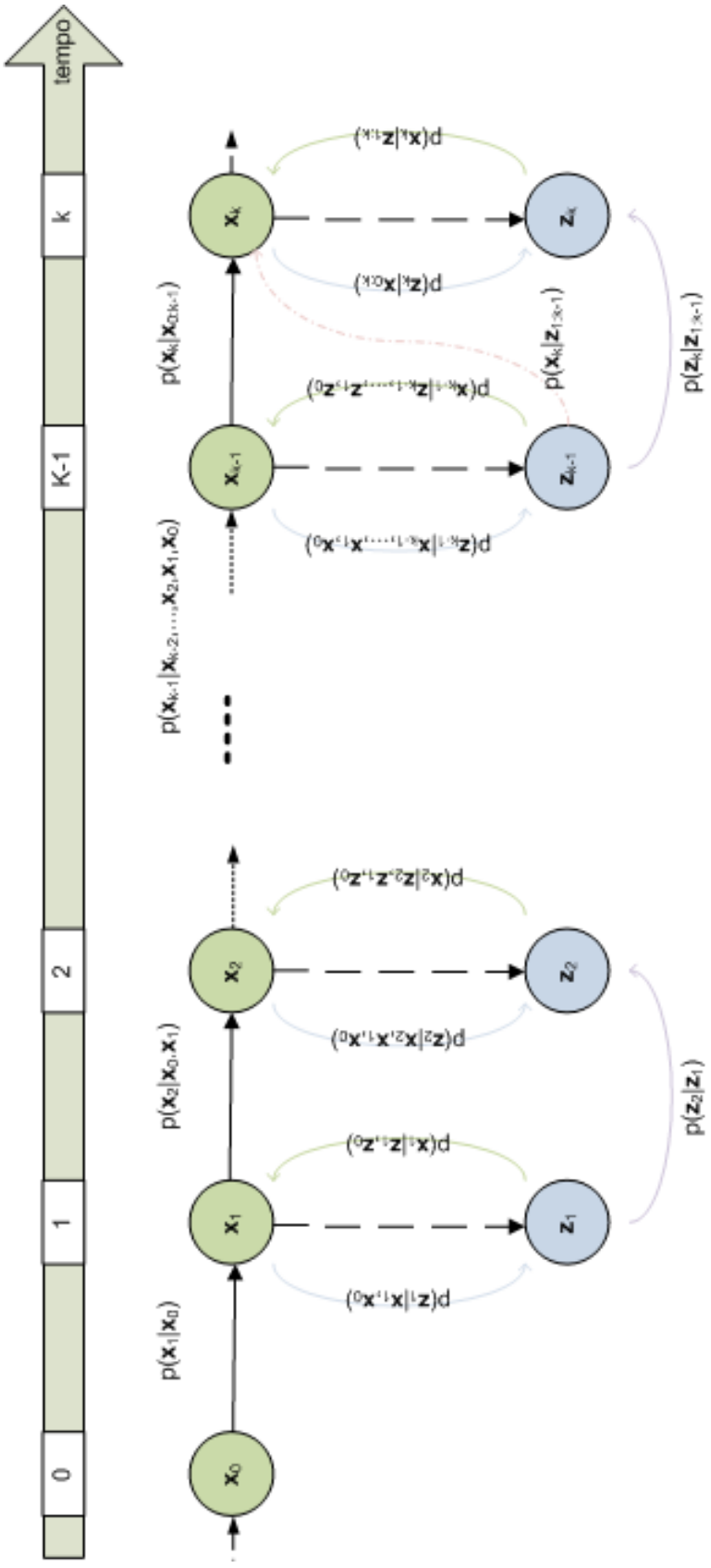

Figura 2.2: Representação gráfica das probabilidades conjuntas em um sistema dinâmico, no tempo discreto k.

ao Teorema de Bayes apresentado na Equação (2.15), acrescido do conjunto de amostras com ruído do sinal, representado pelas observações obtidas do sistema, $\mathbf{z}_{1:k}$, é possível encontrar uma solução algébrica para a probabilidade *a posteriori*, dada por

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})\ p(\mathbf{x}_{0:k})}{p(\mathbf{z}_{1:k})} \quad , \tag{2.15}$$

onde $p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})$ é a densidade de probabilidade conjunta de verossimilhança, $p(\mathbf{x}_{0:k})$ é a densidade de probabilidade *a priori* do sistema dinâmico, ou seja, o conhecimento prévio da dinâmica do sistema alvo, e $p(\mathbf{z}_{1:k})$ é a evidência das amostras de observação do sistema imerso em ruído aditivo branco.

A Equação (2.16) apresenta a evidência após aplicar a definição da probabilidade total [Pap91], que, devido a sua estacionariedade no tempo, representa uma constante de normalização:

$$p(\mathbf{z}_{1:k}) = \int p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})\ p(\mathbf{x}_{0:k})\ d\mathbf{x}_{0:k} \quad . \tag{2.16}$$

Substituindo a Equação (2.16) na Equação (2.15), podemos reescrevê-la como

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})\ p(\mathbf{x}_{0:k})}{\int p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})\ p(\mathbf{x}_{0:k})\ d\mathbf{x}_{0:k}} \quad . \tag{2.17}$$

Restringindo a natureza das amostras de observação como um processo de Markov de primeira ordem, o estado do vetor $\mathbf{x}_k$ é dependente apenas do estado $\mathbf{x}_{k-1}$, eliminando a necessidade de armazenar o conjunto de todas os estados do sistema, $\mathbf{x}_{1:k-1}$. Essa estrutura de processamento, além de ser um modelo natural para fenômenos físicos que ocorrem na natureza, justifica o uso da modelagem temporal do sistema como o espaço de estados permite.

Porém, o cálculo da função de densidade *a posteriori* apresentada na Equação (2.17) exige o conhecimento de todos os estados do vetor de estados do sistema, bem como suas amostras observáveis, o que representa um alto custo para a sua implementação. Uma alternativa é calcular sequencialmente a densidade de probabilidade *a posteriori* marginal $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ através da técnica de predição e atualização (*prediction-update*), restringindo assim o armazenamento dos estados do sistema.

Partindo deste princípio, a forma realizável de construir um processamento da probabilidade marginal *a posteriori* $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ inicia-se com a definição da função de probabilidade conjunta $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ aplicada à definição do Teorema de Bayes, como apresentado

na Equação (2.15). A Equação (2.18) apresenta a expressão final da densidade de probabilidade de verossimilhança $p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})$ em função apenas dos termos da amostra de observação $\mathbf{z}_k$ e do vetor de estado $\mathbf{x}_k$ no tempo k. O detalhamento da Equação (2.18) encontra-se no Apêndice C.1.

$$p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k}) = p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{z}_{1:k-1}|\mathbf{x}_{0:k-1}) \quad . \tag{2.18}$$

Aplicando o mesmo princípio de Bayes, destaca-se o termo *a priori* no tempo k encontrado na Equação (2.15) dos demais termos, como indicado por

$$p(\mathbf{x}_{0:k}) = p(\mathbf{x}_k|\mathbf{x}_{0:k-1})\ p(\mathbf{x}_{0:k-1}) \quad . \tag{2.19}$$

Analogamente, interpretando-se este conceito para a distribuição de probabilidade, a transição de estados de um processo de Markov de primeira ordem, genericamente representada por $p(\mathbf{x}_k|\mathbf{x}_{1:k-1})$, é identificada apenas pela relação entre o vetor de estado nos tempos k e k-1, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, desconsiderando os demais estados, dessa forma é possível considerar que

$$p(\mathbf{x}_k|\mathbf{x}_{1:k-1}) \triangleq p(\mathbf{x}_k|\mathbf{x}_{k-1}) \quad . \tag{2.20}$$

Assim a restrição apresentada na Equação (2.20) aplicada à Equação (2.19), resulta em

$$p(\mathbf{x}_{0:k}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})\ p(\mathbf{x}_{0:k-1}) \quad . \tag{2.21}$$

Por fim, a equação de evidência $p(\mathbf{z}_{1:k})$ pode ser representada por sua forma sequencial

$$p(\mathbf{z}_{1:k}) = p(\mathbf{z}_k|\mathbf{z}_{1:k-1})\ p(\mathbf{z}_{1:k-1}) \quad . \tag{2.22}$$

Aplicando as Equações (2.18), (2.21) e (2.22) na Equação (2.15) é possível repartir a densidade de probabilidade *a posteriori* em duas classes de termos distintos: (i) os termos que representam os elementos obtidos no tempo k, e (ii) o termo que representa o conjunto de elementos acumulados entre o intervalos de tempo entre 0 até k-1, de tal maneira que

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = \frac{[p(\mathbf{z}_{1:k-1}|\mathbf{x}_{0:k-1})\ p(\mathbf{z}_k|\mathbf{x}_k)]\ [p(\mathbf{x}_{0:k-1})\ p(\mathbf{x}_k|\mathbf{x}_{k-1})]}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})\ p(\mathbf{z}_{1:k-1})} \quad . \tag{2.23}$$

Sabendo que a função de densidade *a posteriori* conjunta no tempo k-1 pode ser representada por

$$p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) = \frac{p(\mathbf{z}_{1:k-1}|\mathbf{x}_{0:k-1})\ p(\mathbf{x}_{0:k-1})}{p(\mathbf{z}_{1:k-1})} \quad , \tag{2.24}$$

e substituindo a Equação (2.24) na Equação (2.23), obtém-se

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = \left[\frac{p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})}\right] p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \quad . \tag{2.25}$$

De posse da Equação (2.25), torna-se possível calcular sequencialmente a distribuição de probabilidade *a posteriori* conjunta no tempo k a partir da observação da amostra $\mathbf{z}_k$, do conhecimento das probabilidades *a priori* e da verossimilhança entre a amostra obtida e o espaço de estados $\mathbf{x}_k$, além da distribuição conjunta *a posteriori* obtida em k-1. Entretanto, por envolver o cálculo da distribuição conjunta *a posteriori*, isso exige o conhecimento prévio das distribuições que não são realizáveis numericamente [Can09]. Uma opção é construir uma solução para o cálculo da distribuição marginal *a posteriori* $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ que seja realizável numericamente.

Partindo da equação de predição da probabilidade marginal *a posteriori* $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$, temos conhecimento das amostras até k-1, $\mathbf{z}_{1:k-1}$, além do valor do vetor de estado $\mathbf{x}_{k-1}$. Utilizando a propriedade de regra da cadeia, associada à densidade condicional, é possível chegar à relação

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p[(\mathbf{x}_k|\mathbf{z}_{1:k-1})|(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})]\ p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})\ d\mathbf{x}_{k-1} \quad . \tag{2.26}$$

Realizando as substituições na Equação (2.26) $a = (\mathbf{x}_k|\mathbf{z}_{1:k-1})$ e $b = (\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$, e utilizando a regra da cadeia $p(a,b) = p(b)\ p(b|a)$, obtém-se

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k, \mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})\ d\mathbf{x}_{k-1} \quad . \tag{2.27}$$

Aplicando a equação de Chapman-Kolmogorov [Pap91] e considerando o processo $\mathbf{x}_k$ um processo de Markov, a Equação de predição (2.27) pode ser reescrita por

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\ p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})\ d\mathbf{x}_{k-1} \quad . \tag{2.28}$$

Desta forma é demonstrado que a equação de predição da distribuição de probabilidade *a posteriori* pode ser resolvida numericamente a partir do conhecimento *a priori* do comportamento do processo $\mathbf{x}_k$, além da função de probabilidade *a posteriori* obtida em k-1. Quando é obtida a amostra do vetor de observação no tempo k, $\mathbf{z}_k$, é possível calcular a equação de correção $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. A partir da regra de Bayes [Can09], temos a equação de correção representada por

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{z}_{1:k}) &= \frac{p(\mathbf{x}_k, \mathbf{z}_{1:k})}{p(\mathbf{z}_{1:k})} \\ &= \frac{p(\mathbf{x}_k, \mathbf{z}_k, \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k, \mathbf{z}_{1:k-1})} \quad . \end{aligned} \tag{2.29}$$

Aplicando a regra da cadeia na Equação (2.29), chega-se à relação

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{z}_{1:k-1})\ p(\mathbf{x}_k|\mathbf{z}_{1:k-1})\ p(\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})\ p(\mathbf{z}_{1:k-1})} \quad . \tag{2.30}$$

Devido à independência entre as amostras $\mathbf{z}_k$ e o processo $\mathbf{x}_k$, representado como um processo de Markov, é possível obter a relação $p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{z}_{1:k-1}) = p(\mathbf{z}_k|\mathbf{x}_k)$. Assim, a Equação (2.30) pode ser reescrita pela equação de correção

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \quad , \tag{2.31}$$

onde $p(\mathbf{z}_k|\mathbf{x}_k)$ representa a distribuição marginal de probabilidade de verossimilhança, dada pela observação do modelo proposto; $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ representa a distribuição de probabilidade *a posteriori* obtida a partir da Equação (2.28); e $p(\mathbf{z}_k|\mathbf{z}_{1:k-1})$ representa uma constante de normalização, como representada pela Equação (2.16).

A Tabela 2.1 apresenta o algoritmo consolidado para a estimação Bayesiana sequencial. Devido à integral, necessária para calcular $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$, este termo não é realizável numericamente e requer o uso de uma metodologia empregando integração por Monte Carlo, como será visto na Seção 2.4.

Tabela 2.1: Algoritmo consolidado da estimação bayesiana sequencial.

$k = 0$	Inicialização
	$p(\mathbf{x}_0)$
$@k-1$	conhecido $p(\mathbf{x}_{k-1}\|\mathbf{z}_{1:k-1})$
	predição (não realizável numericamente):
	$p(\mathbf{x}_k\|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k\|\mathbf{x}_{1:k-1})\ p(\mathbf{x}_{k-1}\|\mathbf{z}_{1:k-1})\ d\mathbf{x}_{k-1}$
$@k$	observação $\mathbf{z}_k$ disponível, obtém $p(\mathbf{z}_k\|\mathbf{x}_k)$
	$p(\mathbf{x}_k\|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k\|\mathbf{x}_k)\ p(\mathbf{x}_k\|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k\|\mathbf{z}_{1:k-1})}$
	estima $\hat{\mathbf{x}}_k = E\{\mathbf{x}_k\|\mathbf{z}_{1:k}\}$

Seguindo o espaço de estados descrito pelas Equações (2.13) e (2.14), é possível construir um modelo Bayesiano, a partir das distribuições de probabilidade do sistema dinâmico. É assumido que tanto o vetor de estados $\mathbf{x}_k$, Markoviano, quanto o vetor de observação $\mathbf{z}_k$, são variáveis aleatórias independentes entre si, discretas no tempo. Para encontrar uma estimação da equação de estados é necessário encontrar recursivamente a distribuição de probabilidade *a posteriori* $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ através de dois estágios (*prediction-update*): predição, dado pela Equação (2.28), e correção, definido na Equação (2.31),

acrescido das condições iniciais das equações de estados $p(\mathbf{x}_0)$ e da função de distribuição de probabilidade *a priori* da transição do vetor de estados $\mathbf{x}_{k-1} \Rightarrow \mathbf{x}_k$, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. Deve-se observar que este modelo utiliza o conjunto de todas as observações disponíveis do sistema até o tempo k, definido por $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\}$.

A Figura 2.3 apresenta o modelo proposto para o sistema dinâmico considerando o vetor de estados como um processo de Markov. Se comparado com o modelo geral apresentado pela Figura 2.2, é possível identificar que as probabilidades *a priori* de $\mathbf{x}_k$ não mais dependem do conjunto de todos os estados do sistema, $\mathbf{x}_{0:k-1}$, mas apenas de seu estado imediatamente anterior $\mathbf{x}_{k-1}$.

Para se chegar a essa conclusão sobre a estimativa de estados, partimos da definição de estimação ótima que procura minimizar o erro médio quadrático calculado a partir da diferença entre o vetor de estados real do sistema $\mathbf{x}_k$ e a sua estimação $\hat{\mathbf{x}}_k$, dado o conjunto de observações $\mathbf{z}_{1:k}$ realizados no sistema.

O erro médio quadrático Bayesiano pode ser definido como o valor médio quadrático do erro entre o vetor estimado $\hat{\mathbf{x}}_k$ e o valor real de $\mathbf{x}_k$, como apresentado por [Kay93]

$$B_{mse}(\hat{\mathbf{x}}_k) = E\{(\mathbf{x}_k - \hat{\mathbf{x}}_k)^2\} \quad . \tag{2.32}$$

Como $\mathbf{x}_k$ é um vetor de variáveis aleatórias, o valor médio está associado à distribuição conjunta $p(\mathbf{z}_k, \mathbf{x}_k)$. Assim teremos o valor de erro mínimo quadrático Bayesiano definido pela integral dupla

$$B_{mse}(\hat{\mathbf{x}}_k) = \int\int (\mathbf{x}_k - \hat{\mathbf{x}}_k)^2 \; p(\mathbf{z}_k, \mathbf{x}_k) \; d\mathbf{z}_k \; d\mathbf{x}_k \quad . \tag{2.33}$$

Sabendo-se que a probabilidade conjunta $p(\mathbf{z}_k, \mathbf{x}_k)$ pode ser expressa por

$$p(\mathbf{z}_k, \mathbf{x}_k) = p(\mathbf{x}_k|\mathbf{z}_k) \; p(\mathbf{z}_k) \quad . \tag{2.34}$$

Substituindo (2.34) em (2.33), é possível separar o termo dependente apenas em função de $\mathbf{x}_k$, conforme

$$B_{mse}(\hat{\mathbf{x}}_k) = \int \Big[\int (\mathbf{x}_k - \hat{\mathbf{x}}_k)^2 \; p(\mathbf{x}_k|\mathbf{z}_k) \; d\mathbf{x}_k \Big] p(\mathbf{z}_k) \; d\mathbf{z}_k \quad . \tag{2.35}$$

Para encontrar uma estimação de $\mathbf{z}_k$ que minimize o erro médio quadrático ($B_{mse}(\hat{\mathbf{z}}_k)$), basta minimizar o termo entre colchetes da Equação (2.35). Calculando a derivada parcial deste termo em relação à $\mathbf{z}$, obtém-se

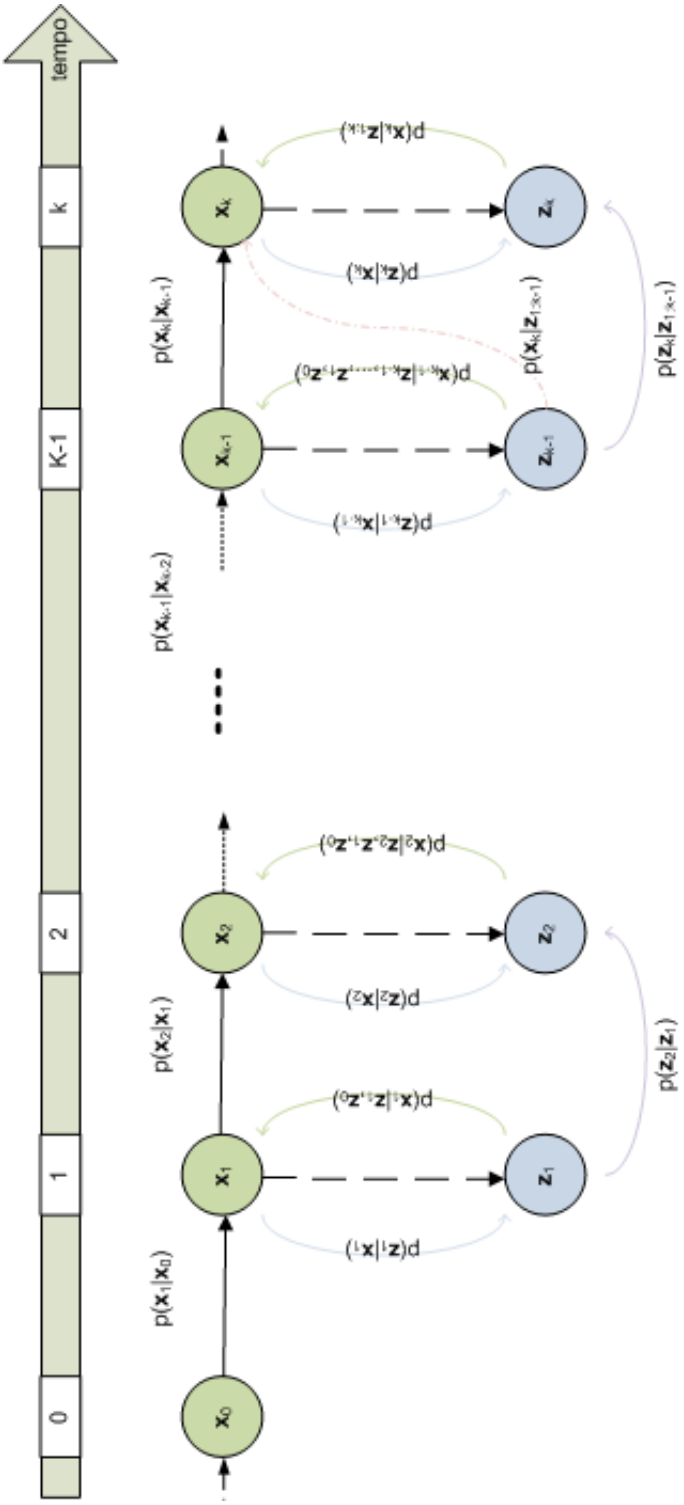

Figura 2.3: Representação gráfica das probabilidades conjuntas em um sistema dinâmico de Markov, no tempo discreto.

$$\begin{aligned}\frac{\partial}{\partial \mathbf{z}_k}\int (\mathbf{x}_k - \hat{\mathbf{x}}_k)^2\, p(\mathbf{x}_k|\mathbf{z}_k)\, d\mathbf{x}_k &= \int \frac{\partial}{\partial \mathbf{z}_k}(\mathbf{x}_k - \hat{\mathbf{x}}_k)^2\, p(\mathbf{x}_k|\mathbf{z}_k)\, d\mathbf{x}_k \\ &= \int -2(\mathbf{x}_k - \hat{\mathbf{x}}_k)\, p(\mathbf{x}_k|\mathbf{z}_k)\, d\mathbf{x}_k \\ &= -2\int \mathbf{x}_k\, p(\mathbf{x}_k|\mathbf{z}_k)\, d\mathbf{x}_k + 2\hat{\mathbf{x}}_k \int p(\mathbf{x}_k|\mathbf{z}_k)\, d\mathbf{x}_k \quad .\end{aligned} \tag{2.36}$$

Considerando que o valor mínimo ocorre quando a relação apresentada na Equação (2.36) é igual a zero:

$$\frac{\partial}{\partial \mathbf{z}_k}\int (\mathbf{x}_k - \hat{\mathbf{x}}_k)^2\, p(\mathbf{x}_k|\mathbf{z}_k)\, d\mathbf{x}_k = 0 \quad . \tag{2.37}$$

Substituindo o termo da esquerda na Equação (2.37) pela Equação (2.36), teremos:

$$\begin{aligned}-2\int \mathbf{x}_k\, p(\mathbf{x}_k|\mathbf{z}_k)\, d\mathbf{x}_k + 2\hat{\mathbf{z}}_k \int p(\mathbf{x}_k|\mathbf{z}_k)\, d\mathbf{x}_k = 0 \\ \hat{\mathbf{x}}_k = \frac{\int \mathbf{x}_k\, p(\mathbf{x}_k|\mathbf{z}_k)\, d\mathbf{x}_k}{\int p(\mathbf{x}_k|\mathbf{z}_k)\, d\mathbf{x}_k} \quad .\end{aligned} \tag{2.38}$$

Como o valor do termo denominador da Equação (2.38) representa a integração $\int p(\mathbf{x}_k|\mathbf{z}_k)d\mathbf{x}_k = 1$, ela pode ser simplificada como

$$\hat{\mathbf{x}}_k = \int \mathbf{x}_k\, p(\mathbf{x}_k|\mathbf{z}_k)\, d\mathbf{x}_k \quad . \tag{2.39}$$

Como apresentado, o estimador ótimo em termos da estimação Bayesiana é obtido a partir do valor médio da distribuição de probabilidade *a posteriori* dada por $p(\mathbf{x}_k|\mathbf{z}_k)$. Logo a Equação (2.39) pode ser reescrita por

$$\hat{\mathbf{x}}_k = E\{\mathbf{x}_k|\mathbf{z}_k\} \quad . \tag{2.40}$$

Fica provado, então, que, encontrando o valor da Equação (2.40), é possível calcular o estimador ótimo através da ótica Bayesiana, $p(\mathbf{x}_k|\mathbf{z}_k)$. Assim, o problema de acompanhamento pode ser calculado recursivamente, com algum grau de incerteza controlado pelo conhecimento estatístico do processo dinâmico, através de duas etapas sucessivas: predição (*prediction*) e correção (*update*). A partir do conhecimento prévio da densidade de probabilidade do estado inicial $p(\mathbf{x}_0) \equiv p(\mathbf{x}_0, \mathbf{z}_0)$, conhecida como a distribuição *a priori*, e definindo a correlação entre as equações de estados apresentados em (2.13) e (2.14) e suas respectivas distribuições de probabilidades como definido nas Equações (2.41) e (2.42):

$$\begin{array}{rcl} \text{equação do sistema} & \Leftrightarrow & \text{fdp de transição} \\ \mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) & \Leftrightarrow & p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) \end{array} \tag{2.41}$$

$$\begin{array}{rcl} \text{equação de observação} & \Leftrightarrow & \text{fdp de verossimilhança} \\ \mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) & \Leftrightarrow & p(\mathbf{z}_k \mid \mathbf{x}_k) \end{array} \tag{2.42}$$

2.4 Método Monte Carlo sequencial

Neste seção é apresentada uma metodologia geral para o desenvolvimento de filtros baseados em Monte Carlo Sequencial (*Sequential Monte Carlo* - SMC) [Can09] onde se utiliza a técnica para calcular uma aproximação da densidade de probabilidade do espaço de estados de um sistema dinâmico, a partir das amostras da sequência de observação obtidas sequencialmente no tempo, ambas tratadas como variáveis aleatórias. Como observado por [DJ09], o método Monte Carlo Sequencial representa o conjunto de ferramentas utilizadas pela metodologia *Particle Filter*. Portanto, são temas distintos, mas que muitas vezes são colocados como a mesma metodologia.

2.4.1 Importance sampling

Dado o conjunto de densidade de probabilidade $\{\pi_k(\mathbf{x}_{0:k})\}$ que representa a variável aleatória $\mathbf{X} \in \mathcal{X}^n$, definindo numa região $\mathcal{D}$ no espaço $\mathcal{X}$, n-dimensional. Para cada ocorrência de $\mathbf{X}$, tal que $\{\mathbf{X} = \mathbf{x}_k, k = 0, 1, \ldots, K\}$, é possível definir sua distribuição de probabilidade $\pi_k(\mathbf{x}_{0:k})$, a partir de [Pap91]

$$\pi_k(\mathbf{x}_{0:k}) = \frac{h_k(\mathbf{x}_{0:k})}{Z_k} \quad , \tag{2.43}$$

onde $h_k(x_{1:k})$ representa uma função de todos os valores do conjunto X_k, enquanto Z_k representa uma constante de normalização, tal que

$$Z_k = \int h_k(\mathbf{x}_{0:k})\, d\mathbf{x}_{0:k} \quad . \tag{2.44}$$

O método Monte Carlo sequencial busca encontrar sequencialmente uma estimativa para $\pi_k(x_{1:k})$, onde k representa intervalos discretos no tempo [DJ09]. Assim, no tempo

k = 1, obtém-se uma aproximação de $\pi_1(x_1)$ e uma estimativa de Z_1, no tempo k = 2, obtém-se uma aproximação de $\pi_2(x_{1:2})$ e uma estimativa de Z_2, e, assim, sucessivamente.

Tendo como base esta metodologia, é proposta desta dissertação aplicá-la no contexto da filtragem digital, onde se deseja encontrar uma estimativa para o vetor de estados $\mathbf{x}_k$, conhecido o conjunto de observações $\mathbf{z}_{1:k}$. Assim, como foi visto na Seção 2.3, a equação de densidade conjunta da probabilidade condicional *a posteriori* é definida por

$$p_{X|Z}(\mathbf{x}|\mathbf{z}) = \frac{p_{X,Z}(\mathbf{x}_{1:k}, \mathbf{z}_{1:k})}{p_Z(\mathbf{z}_{1:k})} \quad . \tag{2.45}$$

É possível aplicar a metodologia SMC no problema de estimação substituindo os elementos desta equação pelos seguintes componentes encontrados num sistema dinâmico modelado a partir da Filtragem Bayesiana, assim como foi apresentado na Equação (2.15) e (2.16):

$$\begin{aligned} h_k(\mathbf{x}_{0:k}) &= p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}) \quad , \\ Z_k &= p(\mathbf{z}_{1:k}) \quad , \\ \pi_k(\mathbf{x}_{0:k}) &= p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \quad . \end{aligned}$$

Assim, consegue-se calcular uma aproximação de $\pi_k(\mathbf{x}_{1:n})$, $\hat{\pi}_k(\mathbf{x}_{1:n})$, definida na Equação (2.43), calculando-se N amostras aleatórias e independentes $X_k^i \backsim \pi_k(x_{0:k})$, $i = 1, \cdots, N$, tal que a aproximação de $\pi_k(\mathbf{x}_{0:k})$ possa ser definida como o valor médio das amostras aleatórias, dada por

$$\hat{\pi}_n(x_k) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_k^i}(x_k) \quad . \tag{2.46}$$

A média I, também conhecida como momento de primeira ordem, da função h(x) pode ser definida por

$$I = \int_D h(\mathbf{x}_k)\ \pi(\mathbf{x}_k)\ d\mathbf{x}_k \quad . \tag{2.47}$$

Uma estimativa da Equação (2.47) é obtida através de $\hat{\pi}_n(x_k)$ pela relação

$$\begin{aligned} I_k &= \int_D h(\mathbf{x}_k)\ \hat{\pi}_{\mathbf{x}_k}\ d\mathbf{x}_k \quad , \\ I_k &= \frac{1}{N} \sum_{i=1}^{N} h(X_k^i) \quad . \end{aligned} \tag{2.48}$$

Assim, o problema de filtragem através da metodologia Monte Carlo sequencial resume em calcular numericamente a integral

$$I = \int_{\mathcal{D}} h(x)\ \pi(x) dx \quad , \tag{2.49}$$

onde $\pi(x)$ é uma função de densidade de probabilidade qualquer e $h(x) \geqslant 0$ é uma função definida na região $\mathcal{D}$.

Uma possível solução númerica para este problema seria utilizar uma técnica Monte Carlo, como por exemplo a aproximação de Riemann, onde todo o espaço n-dimensional seria amostrado uniformemente. Entretanto, essa e muitas outras técnicas de integração por Monte Carlo aumentam a complexidade computacional à medida que a dimensão do espaço n-dimensional aumenta, podendo tornar inviável uma solução numérica.

A idéia por detrás da técnica de *Importance Sampling* está em utilizar um conjunto de amostras aleatórias não uniforme $x^{(1)}, ..., x^{(m)}$, com densidade de probabilidade conhecida e definida por $q(\mathbf{x})$, que se adapte à função de densidade de probabilidade $\pi(\mathbf{x})$, colocando maior peso (ou massa) nas amostras mais significativas. Na prática, isso significa coletar a maior quantidade de amostras onde à função de distribuição de probabilidades (fdp) possui maior importância ou densidade, diminuindo a complexidade computacional do algoritmo.

A maior dificuldade desta técnica está em encontrar uma distribuição $q(\mathbf{x})$ para os pontos de massas mais relevantes que sejam proporcionais à pdf $\pi(\mathbf{x})$ de tal forma que possa ser calculada numericamente a integral apresentada na Equação (2.47).

Assim, é realizado um conjunto de N amostras aleatórias $X^i_{1:n}$, obtidas com a distribuição de probabilidade definida para $q(\mathbf{x})$. Para cada amostra de $X^i_{0:k}$, $i = 0, 1, \ldots, N$, é calculado um fator de massa $\{w^i_k\}$, não normalizado, que determina a proporção de cada amostra em relação a distribuição de probabilidades original do sistema dinâmico, $\pi_k(\mathbf{x})$, e a distribuição de probabilidade proposta (*proposal distribution*), $g_k(\mathbf{x}_{0:k})$, dado por

$$w_k(\mathbf{x}_{0:k}) = \frac{\pi_k(\mathbf{x}_{0:k})}{g_k(\mathbf{x}_{0:k})} \quad . \tag{2.50}$$

Para se obter o peso normalizado $\{W^i_k\}$, basta dividir cada massa calculada na Equação (2.50) pelo somatório de todos os pesos, conforme

$$W^i_k = \frac{w_k(\mathbf{x}^i_{0:k})}{\sum_{j=1}^{N} w_k(\mathbf{x}^j_{0:k})} \quad . \tag{2.51}$$

Assim, é possível calcular uma aproximação para a integral definida pela Equação (2.47) a partir da aproximação da densidade de probabilidade $\hat{\pi}_k(\mathbf{x}_{0:k})$ e do fator de

normalização Z_k definido a partir das Equaçôes (2.52) e (2.53):

$$\hat{\pi}_k(\mathbf{x}_{0:k}) = \sum_{i=1}^{N} W_k^i \, \delta_{X_{0:k}^i}(\mathbf{x}_{0:k}) \quad , \tag{2.52}$$

$$\hat{Z}_n = \frac{1}{N} \sum_{i=1}^{N} w_n(X_{1:n}^i) \quad , \hat{Z}_n = \frac{1}{N} \sum_{i=1}^{N} w_n(X_{1:n}^i) \quad , \tag{2.53}$$

chegando finalmente à equação aproximada da integração apresentada na Equação (2.47) que pode ser escrita como

$$\begin{aligned} I_k^{IS}(h_k(\mathbf{x})) &= \int h(\mathbf{x}_{0:k}) \, \hat{\pi}_k(\mathbf{x}_{0:k}) \, d\mathbf{x}_{0:k} \quad , \\ I_k^{IS}(h_k(\mathbf{x})) &= \sum_{i=1}^{N} W_k^i \, h_k(X_{0:k}^i) \quad . \end{aligned} \tag{2.54}$$

Este resultado é de fundamental importância para se chegar ao cálculo das partículas (*particles*) geradas pelo método *Particle Filter*, como será visto na Seção 2.5.

2.4.2 *Sequential importance sampling*

Como o algoritmo consolidado do filtro Bayesiano sequencial apresentado na Tabela 2.1 não é realizável na prática, devido ao grande número de amostras que são necessárias para se conseguir calcular as integrais envolvidas, pode-se conseguir uma solução numérica através do método *Importance Sampling*, apresentado na Seção 2.4.1. Entretanto, é necessário implementá-lo em uma ótica sequencial [Can09], considerando assim somente os valores encontrados no tempo imediatamente anterior ao tempo que se deseja calcular a estimativa da distribuição *a posteriori* marginal, $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. Partindo deste princípio, devemos definir uma distribuição proposta (*proposal distribution*), $q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ que aceite uma distribuição marginal

$$q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \, q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) \quad . \tag{2.55}$$

Aplicando a regra da cadeia na Equação (2.55), temos

$$q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = q(\mathbf{x}_0) \, \prod_{i=1}^{k} q(\mathbf{x}_i|\mathbf{x}_{0:k-i}, \mathbf{z}_{1:i}) \quad . \tag{2.56}$$

O peso não normalizado, introduzido pela Equação (2.50), é computado recursivamente aplicando nele a Equação (2.55), obtendo-se

$$\begin{aligned} w_k(\mathbf{x}_{0:k}) &= \frac{p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})\ p(\mathbf{x}_{0:k})}{q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})} \\ &= \frac{p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})\ p(\mathbf{x}_{0:k})}{q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})\ q(\mathbf{x}_k|\mathbf{x}_{0:k-1},\mathbf{z}_{1:k})} \\ &= \frac{p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})\ p(\mathbf{x}_{0:k})}{q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})\ q(\mathbf{x}_k|\mathbf{x}_{0:k-1},\mathbf{z}_{1:k})} \left[\frac{p(\mathbf{z}_{1:k-1}|\mathbf{x}_{0:k-1})\ p(\mathbf{x}_{0:k-1})}{p(\mathbf{z}_{1:k-1}|\mathbf{x}_{0:k-1})\ p(\mathbf{x}_{0:k-1})}\right] \\ &= \left[\frac{p(\mathbf{z}_{1:k-1}|\mathbf{x}_{0:k-1})\ p(\mathbf{x}_{0:k-1})}{q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})}\right] \frac{p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})\ p(\mathbf{x}_{0:k})}{[p(\mathbf{z}_{1:k-1}|\mathbf{x}_{0:k-1})\ p(\mathbf{x}_{0:k-1})]\ q(\mathbf{x}_k|\mathbf{x}_{0:k-1},\mathbf{z}_{1:k})} \quad . \end{aligned} \tag{2.57}$$

A componente em destaque na Equação (2.57) corresponde ao peso não normalizado em $k-1$, ou seja, $w_{k-1}(\mathbf{x}_{0:k-1})$. Na mesma equação, os termos $p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})$ e $p(\mathbf{x}_{0:k})$ podem ser substituídos pelas relações apresentadas nas Equações (2.18) e (2.21), respectivamente, resultando em

$$w_k(\mathbf{x}_{0:k}) = w_{k-1}(\mathbf{x}_{0:k-1}) \left[\frac{p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{z}_{1:k-1}|\mathbf{x}_{0:k-1})}{p(\mathbf{z}_{1:k-1}|\mathbf{x}_{0:k-1})} \frac{p(\mathbf{x}_k|\mathbf{x}_{k-1})\ p(\mathbf{x}_{0:k-1})}{\mathbf{x}_{0:k-1}} \frac{1}{q(\mathbf{x}_k|\mathbf{x}_{0:k-1},\mathbf{z}_{1:k})}\right] . \tag{2.58}$$

Realizando os possíveis cancelamentos na Equação (2.58), chegamos à Equação final, sequencial, para o peso não normalizado

$$w_k(\mathbf{x}_{0:k}) = w_{k-1}(\mathbf{x}_{0:k-1}) \left[\frac{p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{x}_{k-1})}{q(\mathbf{x}_k|\mathbf{x}_{0:k-1},\mathbf{z}_{1:k})}\right] \quad , \tag{2.59}$$

onde podemos identificar o termo calculado em $k-1$ dado por $w_{k-1}(\mathbf{x}_{0:k-1})$ e, em seguida, o termo a ser calculado quando a nova amostra de observação $\mathbf{z}_k$ estiver disponível em k.

Para se obter o peso normalizado W_k no tempo k, basta aplicar a Equação (2.51). Definindo, então, no tempo k, um conjunto de partículas (*particles*), X_k^i, que serão gerados a partir da distribuição de probabilidades $q(\mathbf{x}_k|\mathbf{x}_{0:k-1},\mathbf{z}_{1:k})$, que é conhecida e apresentada por

$$X_k^i = q(\mathbf{x}_k|\mathbf{x}_{0:k-1},\mathbf{z}_{1:k}) \quad . \tag{2.60}$$

O par $\{X_k^i, W_k^i\}$ descreve cada uma da N_p partículas, onde $i = 1,\ldots,N_p$.

A Tabela 2.2 consolida todos os passos necessários para implementar o algoritmo utilizado pelo *Importance Sampling* e, assim, conseguir encontrar uma distribuição marginal para a distribuição de probabilidade *a posteriori* que permita estimar o valor do estado do sistema no tempo k a partir do conjunto de observações obtidas através de $\mathbf{z}_{1:k}$.

Tabela 2.2: Algoritmo consolidado da *Importace Sampling* obtido sequencialmente.

$k = 0$	Inicialização
	amostras iniciais: $X_0^i = q(\mathbf{x}_0)$
	peso não normalizado: $W_0^i = \frac{p(\mathbf{y}_0 \mid X_0^i)\, p(\mathbf{X}_0^i)}{p(\mathbf{x}_0)}$
	peso normalizado: $w_0^i = \sum_{i=1}^{N_p} W_0^i$
$@k-1$	conjunto {amostras, pesos} armazenados: $\{X_{k-1}^i, W_{k-1}^i\}$
$@k$	nova observação disponível: $\mathbf{z}_k$
	novo conjunto de *particles*: $X_k^i \approx q(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})$ com $i = 1, \ldots, N_p$
	atualização dos pesos de cada *particle*: $W_k^i = W_{k-1}^i \frac{p(\mathbf{y}_k \mid X_k^i)\, p(\mathbf{x}_k \mid \mathbf{x}_k^i)}{q(\mathbf{x}_k^i \mid X_{k-1}^i, \mathbf{z}_{1:k})}$
	peso normalizado: $w_k^i = \frac{W_k^i}{\sum_{i=1}^{N_p} W_k^i}$
	cálculo da distribuição *a posteriori* de probabilidade:
	$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \approx \sum_{i=1}^{Np} w_k^i\, \delta(\mathbf{x}_k - \mathbf{x}_k^i)$

2.5 Particle filter

Partindo das demonstrações apresentadas nas Seções 2.2, 2.3 e 2.4, é possível formular a estrutura teórica do *Particle Filter*, centrada no conceito da técnica *Importance Sampling*. O problema de acompanhamento (*tracking*) definido neste trabalho procura estimar a trajetória, em tempo real, de um objeto em uma sequência de vídeo digital. Uma possível abordagem, como apresentado na Seção 2.3, utiliza uma filtragem Bayesiana e emprega um processamento sequencial das evidências observadas quadro a quadro, na tentativa de se estimar estatisticamente seu atual estado (posição e dimensão no tempo k), modelado como um processo Markoviano, causal, discreto no tempo ($k \in \mathbb{N}$). Assim, a metodologia baseada em *Particle Filter* se utiliza dessa perspectiva Bayesiana da modelagem do sistema e, através da técnica de *Importance Sampling*, encontrando uma solução numérica para a construção da densidade de probabilidade *a posteriori*, que é realizada de forma aproximada a partir da integração por simulação Monte Carlo, utilizando um conjunto de N pontos de massa (*particles*) $\{\mathbf{X}_k^i, \omega_k^i\}_{i=1}^{N_p}$.

[KF09] propõe o cenário descrito a seguir para definir as partículas (*particles*).

Considere uma variável aleatória $\mathcal{X}$, representada pela sua realização, o vetor $\mathbf{x}$, tal que $p(\mathbf{x})$ é a sua distribuição de densidade de probabilidade. Deseja-se estimar o valor

esperado da função definida por $f(\mathbf{x})$, tal que

$$E_X\{f(\mathbf{x})\} = \int f(\mathbf{x})\, p(\mathbf{x})\, d\mathbf{x} \quad . \tag{2.61}$$

Pode-se aproximar esta integral através da escolha aleatória de N partículas, de tal modo que a distribuição de probabilidades de $\mathbf{x}$ possa ser descrita como

$$\hat{p}(\mathbf{x}) \approx \frac{1}{N} \sum_{i=1}^{N} \delta(\mathbf{x} - \mathbf{X}^i) \quad . \tag{2.62}$$

Assim, é possível aproximar, pela técnica de Monte Carlo, a estimativa de $f(\mathbf{x})$, Equação (2.61), a partir do seu valor esperado definido por

$$E_X\{f(\mathbf{x})\} = \int f(\mathbf{x})\, \hat{p}(\mathbf{x})\, d\mathbf{x} \approx \frac{1}{N} \sum_{i=1}^{N} \delta(\mathbf{x} - \mathbf{X}^i) \quad . \tag{2.63}$$

Partindo deste princípio de Integração por Monte Carlo, quando a densidade de probabilidade $p(\mathbf{x})$ não é conhecida, ou ao menos é de difícil definição, pode-se empregar a técnica de *Importance Sampling*, descrita na Seção 2.4.1, para se chegar a uma estimação de $E_X\{f(\mathbf{x})\}$, definindo uma nova distribuição $q(\mathbf{x})$ tal que seja possível aleatoriamente amostrar N_p diferentes pontos de massa $\mathbf{X}^i$ e seus respectivos pesos ω^i construindo, para cada tempo k, um conjunto relevante de amostras como apresentado por

$$\{\mathbf{X}^i_{0:k},\, \omega^i_k\}^N_{i=1} \quad . \tag{2.64}$$

Seguindo o cenário proposto neste trabalho, o processo que se deseja estimar é modelado como um processo de Markov de primeira ordem, discreto no tempo e, possivelmente, não linear. Cada amostra em k, pertencente ao conjunto randômico $\mathbf{X}^i_k$ apresentado na Equação (2.60), é gerada a partir da amostra $\mathbf{X}^i_{k-1}$, gerada em $k-1$, além da observação de $\mathbf{z}_k$. Assim, a Equação (2.60) é reescrita em função desses dois valores, como apresentado por

$$X^i_k = q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k) \quad . \tag{2.65}$$

Cada amostra $\mathbf{X}^i_k$ possui um peso w^i_k associado, que representa a sua importância na composição da distribuição de probabilidade marginal *a posteriori* $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. Como apresentado nas Equação (2.59), cada amostra i do conjunto de partículas possui um peso, definido sequencialmente a partir do seu valor em $k-1$, além das probabilidades de verossimelhança $p(\mathbf{z}_k|\mathbf{x}_k)$, da probabilidade de transição de estado $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ e da

distribuição *proposal* $q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})$. Assim, a Equação (2.59) é reescrita para cada elemento do conjunto de partículas

$$\omega_k^i(\mathbf{x}_k) = w_{k-1}^i(\mathbf{x}_{k-1}) \left[\frac{p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{x}_{k-1})}{q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})} \right] \quad , \tag{2.66}$$

onde, $i = 0, 1, \ldots, N_p$ representa cada índice do conjunto de partículas.

Para o caso utilizado pelo *Particle Filter*, cada peso ω_k^i deve ser normalizado, como indicado na Equação (2.51), que pode ser reescrita como

$$W_k^i = \frac{\omega_k^i}{\sum_{j=1}^{N_p}\ \omega_k^i} \quad . \tag{2.67}$$

Como ilustrado na Tabela 2.2, uma aproximação discreta da probabilidade marginal *a posteriori* $p(\mathbf{x}_k|\mathbf{z}_k)$ pode ser obtida pelo somatórios de todos os elementos de massa normalizados

$$p(\mathbf{x}_k|\mathbf{z}_k) \approx \sum_{i=1}^{N_s} W_k^i\, \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad . \tag{2.68}$$

De acordo com as Equações (2.65), (2.66) e (2.68), é possível construir sequencialmente um *Particle Filter* teórico, baseado no conceito de *Importance Sampling*, assim como em [AMGC02]. A Tabela 2.3 apresenta o algoritmo utilizado para implementar esta metodologia de filtragem bayesiana.

Apesar do método *Particle Filter*, quando comparado com outras técnicas de acompanhamento, ter a vantagem de permitir sua implementação num ambiente de processamento multitarefa, onde cada partícula pode ser considerada uma instância independente de processamento, esta abordagem remete ao problema conhecido como degeneração de algumas das partículas do filtro [IB98]. Este fenômeno ocorre devido a um aumento da variância entre as partículas a cada intervalo de tempo, permitindo que um subconjunto de todas as partículas sejam responsáveis pelo cálculo da distribuição de probabilidade *a posteriori*. Assim, um significativo processamento por parte do filtro é realizado por partículas $\mathbf{X}_k^i$ que possuem peso $\mathbf{W}_k^i \to 0$, contribuindo de forma insignificante para o cálculo da probabilidade *a posteriori*, o que torna a filtragem ineficiente apesar da mesma complexidade computacional inicial, quando todas as partículas contribuíam equiprovavelmente.

A Figura 2.4 ilustra qualitativamente este fenômeno. Somente as partículas centrais, em vermelho, possuem pesos significativos para o cálculo da distribuição de probabilidade *a posteriori* do sistema dinâmico, responsáveis consequentemente pela estimação

do estado do processo. As partículas localizadas nos extremos do eixo horizontal, em azul, apesar de exigirem a mesma parcela de processamento do filtro digital, pouco contribuem para o resultados final.

Para identificar o problema de degeneração, [LC98] introduziu em seu trabalho uma métrica chamada *effective sample size* N_{eff}, definida por

$$N_{eff} = \frac{N_p}{1 + Var(\omega_k^{*i})} \quad , \tag{2.69}$$

onde

$$\omega_k^{*i} = \frac{p(\mathbf{x}_k^i|\mathbf{z}_{1:k})}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)} \ .$$

Tabela 2.3: Algoritmo *Particle Filter* Sequencial (SIS).

$k = 0$ (Inicialização)

N_p: Define o número total de amostras utilizadas pelo filtro.

O estado inicial de x_0 é conhecido e possui uma distribuição

de probabilidade definida por $p(\mathbf{x}_0)$.

FOR $i = 1 : N_p$

$\mathbf{X}^i_{k=0} \leftarrow p(\mathbf{x}_0)$

$W^i_{k=0} = \frac{1}{N_p}$

END-FOR

LOOP

$\mathbf{z}_k$: Nova observação é obtida.

Dado o modelo de espaço de estados na ótica Bayesiana,

Equações (2.41) e (2.42).

Define a distribuição *proposal* $q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)$

e aleatoriamente calcule as partículas $\mathbf{x}_k$ sequencialmente.

FOR $i = 1 : N_p$

$\mathbf{X}^i_k \leftarrow q(\mathbf{x}_k|\mathbf{X}^i_{k-1}, \mathbf{z}_k)$

$\omega^i_k \leftarrow \omega^i_{k-1} \left[\frac{p(\mathbf{z}_k|\mathbf{X}^i_k)\, p(\mathbf{X}^i_k|\mathbf{X}^i_{k-1})}{q(\mathbf{X}^i_k|\mathbf{X}^i_{k-1}, \mathbf{z}_k)} \right]$

END-FOR

Normaliza os pesos de massa

FOR $i = 1 : N_p$

$W^i_k \leftarrow \frac{\omega^i_k}{\sum_{i=1}^{N_p} \omega^i_k}$

END-FOR

Estima o valor $\mathbf{x}_k$ a partir da probabilidade marginal *a posteriori* $p(\mathbf{x}_k|\mathbf{z}_k)$

$\hat{p}(\mathbf{x}_k|\mathbf{z}_k) = \sum_{i=1}^{N_p} W^i_k\, \delta(\mathbf{x}_k - \mathbf{X}^i_k)$

$\hat{\mathbf{x}} = E\{\hat{p}(\mathbf{x}_k|\mathbf{z}_k)\}$

$k \leftarrow k + 1$

END-LOOP

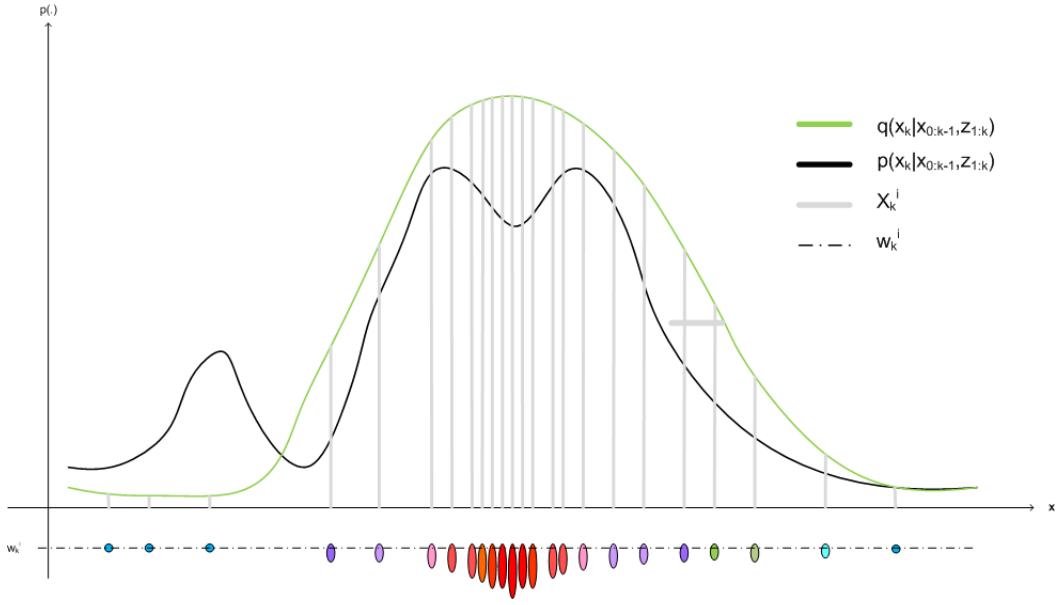

Figura 2.4: Ilustração qualitativa da distribuição *proposal*, utilizada para gerar as partículas. As partículas em cor vermelha são aquelas com maior peso e, portanto, contribuem significativamente com a estimativa do vetor de estados. Por outro lado, as partículas em cor azul, com pesos próximo de zero, têm pouca influência na estimação, muito embora o esforço computacional seja o mesmo para todas as partículas.

N_{eff} não pode ser calculada numericamente. Porém, pode ser calculado sua estimativa $\hat{N}_{eff}$ como

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_p} (W_k^i)^2} \quad , \tag{2.70}$$

onde W_k^i, Equação (2.51), representa o peso normalizado da i-ésima partícula $\mathbf{X}_k^i$ no tempo k.

Quando $N_{eff} \ll N_p$, isso indica que o filtro sofre do problema de degeneração por parte de algumas de suas partículas.

Na literatura especializada sobre este tema, [AMGC02] apresenta duas soluções para dirimir o problema de forma eficiente, além da possibilidade de uma solução por força bruta, em que se utiliza um número muito grande de partículas, $N_p \to \infty$, o que torna impraticável sua implementação em tempo-real.

A primeira tentativa é escolher uma distribuição proposta $q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$ que minimize $Var(\omega_k^{*i})$. Esta metodologia torna a escolha de $q(.)$ fator decisivo para o sucesso do filtro digital. Porém, a mesma está associada ao conhecimento *a priori* do sistema $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. No caso específico deste trabalho, o conhecimento *a priori* não é relevante para o cálculo da estimativa de movimento (como será visto no Capítulo 4). Portanto, esta metodologia não foi considerada na elaboração deste trabalho. Desse modo, a segunda solução proposta, conhecida como *resampling* foi utilizada.

O *resampling* busca substituir, no processamento do filtro, as partículas de pouca relevância, que são aquelas que possuem peso W_k^i muito pequeno e que contribuem para o problema da degeneração, por novas partículas, escolhidas a partir do conjunto que possuem pesos mais significativos. A técnica mais utilizada para a escolha do novo estado da partícula a ser reconfigurada (*resampled*) considera escolher, com probabilidade uniforme $\mathbb{U}[a, b]$, o espaço onde as partículas com pesos relevantes estão concentradas.

A Figura 2.5 apresenta uma ilustração qualitativa da proposta *resampling* para o caso de um sistema de uma dimensão, $X_k \in \mathbb{R}$. Na ilustração, existem três partículas, identificadas por $\mathbf{X}_k^{\xi^1}$, $\mathbf{X}_k^{\xi^2}$ e $\mathbf{X}_k^{\xi^3}$, com seus respectivos pesos, $W_k^{\xi^1}$, $W_k^{\xi^2}$ e $W_k^{\xi^3}$, desprezíveis se comparados com o restante das partículas do filtro, e que pouco contribuem na construção da distribuição de probabilidade *a posteriori*. Dessa forma, essas partículas são elegíveis a serem reconfiguradas, considerando o novo estado $\mathbf{X}_k^{\xi}$ no intervalo compreendido entre as extremidades **a** e **b**, com probabilidade uniforme dentro deste intervalo.

A Tabela 2.4 apresenta o algoritmo genérico para o método *Particle Filter*, pre-

vendo o problema de *resampling* e, assim, definindo um limite inferior N_T para $\hat{N}_{eff}$. Caso $\hat{N}_{eff} < N_T$, é aplicado o processamento de *resampling* ao filtro digital, antes de iniciar uma nova observação do sistema $\mathbf{z}_k$.

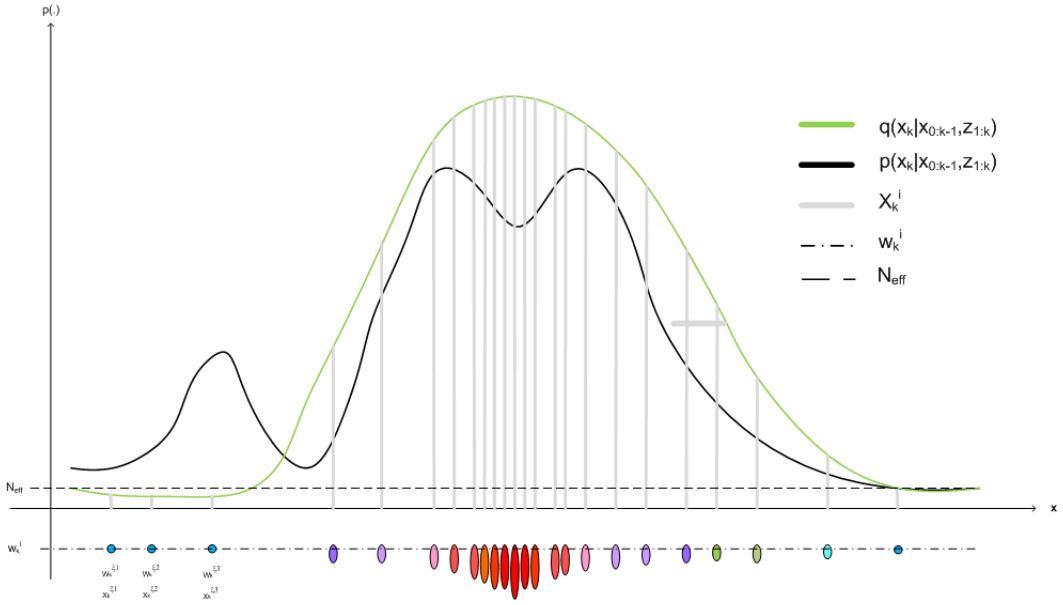

Figura 2.5: As partículas abaixo do valor calculado para *effective sample size* serão reamostradas, devido à sua pouca contribuição ao processamento do *Particle Filter* para a estimação do vetor de estado do sistema.

Tabela 2.4: Algoritmo geral para o método SIS com *resampling*.

$k = 0$ (Inicialização)

$N_p,\ N_T$: Define o número total de amostras utilizadas pelo filtro. Valor mínimo para $\hat{N}_{eff}$.

O estado inicial de x_0 é conhecido e possui uma distribuição

de probabilidade definida por $p(\mathbf{x}_0)$.

FOR $i = 1 : N_p$

$\mathbf{X}^i_{k=0} \leftarrow p(\mathbf{x}_0)$

$W^i_{k=0} = \frac{1}{N_p}$

END-FOR

LOOP

$\mathbf{z}_k$: Nova observação é obtida.

Dado o modelo de espaço de estados, Equações (2.41) e (2.42).

Define a distribuição *proposal* $q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)$

e aleatoriamente calcule as partículas $\mathbf{x}_k$ sequencialmente.

FOR $i = 1 : N_p$

$\mathbf{X}^i_k \leftarrow q(\mathbf{x}_k|\mathbf{X}^i_{k-1}, \mathbf{z}_k)$

$\omega^i_k \leftarrow \omega^i_{k-1} \left[\frac{p(\mathbf{z}_k|\mathbf{X}^i_k)\ p(\mathbf{X}^i_k|\mathbf{X}^i_{k-1})}{q(\mathbf{X}^i_k|\mathbf{X}^i_{k-1}, \mathbf{z}_k)} \right]$

END-FOR

Normaliza os pesos de massa

FOR $i = 1 : N_p$

$W^i_k \leftarrow \frac{\omega^i_k}{\sum_{i=1}^{N_p} \omega^i_k}$

END-FOR

Calcula $\hat{N}_{eff}$: $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_p} (W^i_k)^2}$

Estima o valor $\mathbf{x}_k$ a partir da probabilidade marginal *a posteriori* $p(\mathbf{x}_k|\mathbf{z}_k)$

$\hat{p}(\mathbf{x}_k|\mathbf{z}_k) = \sum_{i=1}^{N_p} W^i_k\ \delta(\mathbf{x}_k - \mathbf{X}^i_k)$

$\hat{\mathbf{x}} = E\{\hat{p}(\mathbf{x}_k|\mathbf{z}_k)\}$

IF $\hat{N}_{eff} < N_T$ THEN Aplicar *resampling*

$k \leftarrow k + 1$

END-LOOP

Capítulo 3

Modelagem da face

3.1 Introdução

A face do interlocutor é o objeto que se deseja acompanhar durante a estabilização do vídeo. Para tornar o acompanhamento robusto é preciso definir um modelo para a face, de modo que o mesmo possibilite identificá-la a cada novo quadro de vídeo, evitando assim considerar outros objetos da cena que não possuem a mesma forma e cor. Uma elipse parametrizada é utilizada como modelo para a face como apresentado a seguir, seguindo o mesmo conceito desenvolvido em [Bir98].

3.2 Vetor de estados modelado como uma elipse parametrizada

Assumindo uma sequência de vídeo contendo três componentes de cor ou planos (R, G, B) [GW08], ortogonalmente amostrado pela matriz

$$\mathbf{V} = \begin{bmatrix} \Delta x_1 & 0 & 0 \\ 0 & \Delta x_2 & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \quad . \tag{3.1}$$

As componentes $\{R, G, B = 0, 1, 2, \ldots, 255\}$ representam, respectivamente, os canais vermelho, verde, e azul, e formam o modelo de cor utilizado neste trabalho.

Um quadro do vídeo no intervalo de tempo discreto k pode ser considerado como uma imagem bidimensional, representada por uma matriz 2-D de dimensões **m** por **n**,

onde **m** é a dimensão do eixo horizontal, enquanto **n** é a dimensão do eixo vertical. Cada elemento da matriz $\mathbf{m} \times \mathbf{n}$ possui um valor inteiro de intensidade, conhecido como *pixel*. Assim $\mathbf{z}_k^{(n)}(\mathbf{x})$ é a representação utilizada nesta dissertação para referir-se a cada elemento do quadro de vídeo digital (*pixel*) de n planos no intervalo de tempo discreto k, onde $\{n > 0,\ n \in \mathbb{N}\}$ é o número de planos amostrados em cada quadro, $\{k \geq 0,\ k \in \mathbb{N}\}$ é o intervalo de tempo discreto e $\mathbf{x} = [x_1 \quad x_2]$ é o vetor de coordenadas espaciais com $\{x_1 > 0,\ x_1 \in \mathbb{N}\}$, sendo a componente horizontal, enquanto $\{x_2 > 0,\ x_2 \in \mathbb{N}\}$ é a componente vertical.

É assumido que cada quadro do vídeo digital corresponde a uma amostra no tempo k, representado por $\mathbf{z}_k$. No instante inicial k_0, a face do interlocutor a ser acompanhada pelo algoritmo é modelada como uma elipse com o centro nas coordenadas (x_0, y_0), os eixos a_0 e b_0, representando o eixo menor (vertical) e maior (horizontal) da face respectivamente e ϕ_0, o deslocamento angular da face em relação aos eixos horizontal e vertical. Para cada novo quadro, a face pode deslocar-se, rotacionar-se ou mudar sua escala. Essas três possíveis transformações geométricas são conhecidas como: translação, rotação e escalonamento.

Para obter uma equação implícita da elipse, basta considerar o caso geral de uma superfície cônica, que é representada pelo polinômio de segunda ordem

$$F(\mathbf{a}; \mathbf{x}) = \mathbf{a} \cdot \mathbf{x} = ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad , \tag{3.2}$$

onde $\mathbf{a} = [a\ b\ c\ d\ e\ f]^T$ representa os parâmetros implícitos de uma superfície cônica e $\mathbf{x} = [x^2\ xy\ y^2\ x\ y\ 1]^T$. A elipse será definida quando a base da superfície cônica estiver a condição $b^2 - 4\,a\,c < 0$ for atendida.

A Equação (3.2) identifica a posição dos pontos x e y que formam o contorno da elipse no espaço vetorial. Em um problema de estimação dos parâmetros da elipse, a partir de um conjunto de n-pontos pertencentes a este contorno, é adequado empregar uma técnica *Least-Square* (LS), como proposto em [FPF99]. Esta abordagem será utilizada neste trabalho para definir a região de contorno da face. Essa técnica busca minimizar a soma quadrática das distâncias dos n-pontos escolhidos com o melhor modelo prosposto para a elipse que atenda a condição

$$\mathcal{D}_{\mathcal{A}}(a) = \sum_{i=1}^{N} F(\mathbf{x}_i)^2 \quad , \tag{3.3}$$

onde $\{x_i,\ i = 1, \ldots, N\}$ representa o vetor de coordenadas e $\{\mathbf{a} \in \mathcal{A}\}$ é o vetor de parâmetros que se deseja estimar.

A modelagem da face proposta neste trabalho utiliza a forma canônica para representar a elipse, dada por

$$\frac{(x - x_c)^2}{a^2} + \frac{(y - y_c)^2}{b^2} = 1 \quad , \tag{3.4}$$

onde (x_c, y_c) é o centro da elipse no eixo de coordenadas XY do espaço vetorial $\mathcal{R}^2$, enquanto a e b são, respectivamente, os eixos menor e maior da elipse.

Assim o vetor de estados $\mathbf{x}_k$ que representa a face que atende a condição definida pela Equação (3.3), no instante de tempo k, é definido por

$$\mathbf{x} = [x_c\ y_c\ a\ b\ \phi]^T \quad , \tag{3.5}$$

onde ϕ é o desvio angular, medido em radianos dos eixos a e b a partir de algum referência, como, por exemplo, o eixo de coordenadas horizontal. A Figura [3.1] ilustra os parâmetros da elipse descrita.

[Bir98] concluiu que a cabeça humana pode ser modelada por uma elipse em que a relação entre o eixo maior b e o eixo menor a corresponde a uma razão fixa de $1,2$ ($\frac{b}{a} = 1,2$). Considerando válida esta afirmação, é possível reduzir a dimensão do vetor de espaço de estados $\mathbf{x}$ de cinco para quatro parâmetros, $\{\mathbf{x} \in \mathcal{R}^4\}$ e, consequentemente, o custo de processamento para o cálculo da estimação.

Para possibilitar encontrar qual o deslocamento da face quadro a quadro é preciso calcular o vetor de movimento da face, que é definido como o vetor diferença entre a posição da face no quadro $\mathbf{z}_{k-1}$ e o quadro $\mathbf{z}_k$, o que é representado por

$$\Delta\mathbf{x}_k = \mathbf{x}_k - \mathbf{x}_{k-1} \quad . \tag{3.6}$$

Para o cálculo do deslocamento, é preciso apenas a informação referente às coordenadas x e y do centro da elipse. Como a Equação (3.6) traduz todas as transformações geométricas permitidas, pode-se representar apenas as informações das coordenadas citadas, através da equação simplificada

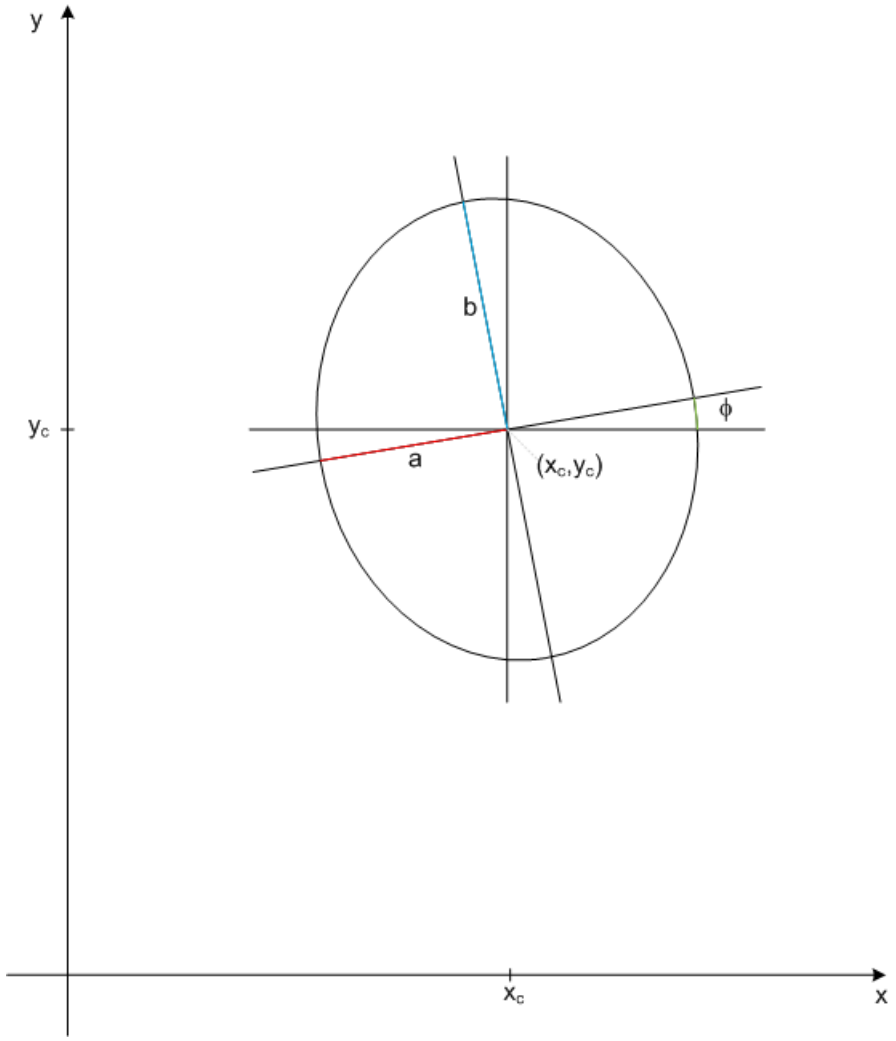

Figura 3.1: Parâmetros da elipse na forma canônica. Os eixos menor e maior da elipse são representados por **a** e **b**. ϕ representa o deslocamento angular em relação às coordenadas cartesianas e, por fim, o centro da elipse é representado por (x_c, y_c).

$$\Delta \mathbf{x}_k = \begin{bmatrix} (x_c(k) - x_c(k-1)) \\ (y_c(k) - y_c(k-1)) \\ 0 \\ 0 \end{bmatrix} \quad . \tag{3.7}$$

Isso permite criar um modelo de primeira ordem para o movimento da face quadro a quadro, dado por

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta \mathbf{x}_k + \upsilon_k \quad , \tag{3.8}$$

onde υ_k representa o erro de estimação da posição corrente da face, no tempo k, que pode

ser modelado como um ruído branco Gaussiano de média zero, definido por

$$\upsilon_k = N(0, \Sigma_G) \quad . \tag{3.9}$$

3.3 Técnica *least-squares* (LS) para definição do contorno da face

Como será apresentado no Capítulo 4, a técnica identificada nesta dissertação como *Método dos Blocos Particionados* (BP) é empregada a cada quadro do vídeo para identificar as regiões no quadro $\mathbf{z}_k$ onde provavelmente encontra-se localizada a face do interlocutor.

É preciso definir uma estimativa inicial, mesmo que imprecisa, da posição e do tamanho da face identificada pelo vetor de estados $\mathbf{x}$ apresentado na Equação (3.5), a partir do modelo apresentado na Seção 3.2. O *Método dos Blocos Particionados* retorna somente um conjunto de pontos no quadro s_k que podem ser considerados como a região de contorno da face. A técnica apresentada nesta seção realiza uma transformação desses pontos de contorno no modelo parametrizado da elipse, permitindo, assim, calcular uma estimativa do centro da face, bem como dos parâmetros em forma canônica clássica da elipse que a modela, identificado como $\hat{x}_k$. A técnica *least-squares* apresentada por [FPF99], conhecida como *Direct Least Squares*, é utilizada nesta etapa como apresentada nesta seção.

A partir de um modelo previamente definido, é possível empregar em geral uma técnica LS que encontre o conjunto de parâmetros que represente este modelo, a partir de um critério de distância mínima quadrática, com o foco em estimar a melhor elipse que represente o conjunto de pontos identificados como a região de contorno da face.

Partindo do polinômio de segunda ordem que generaliza a superfície cônica num plano $\mathbb{R}^3$, como apresentado na Equação (3.2), e aplicando a restrição $4\,a\,c - b^2 = 1$, é possível derivar uma elipse [FPF99] no contorno da superfície e, partindo dessa condição, poder encontrar, a partir de um conjunto conhecido de pontos, os parâmetros de uma elipse utilizando um estimador do tipo LS.

O método LS foi aplicado neste trabalho na intenção de encontrar os parâmetros da elipse que formará o contorno do objeto a ser acompanhado, no caso, a face do interlocutor

na sequência de vídeo digital. O critério empregado neste método é definir uma função $J(\omega)$ que minimize a soma quadrática de erro da função $F(\omega)$ qualquer, definida como

$$J(\omega) = \sum_{i=1}^{N} F(\mathbf{x}_i)^2 \quad , \tag{3.10}$$

onde a função $F(\mathbf{x}_i)$ representa a métrica adotada, a ser otimizada em relação ao parâmetro desconhecido ω, que pode ser um conjunto de dados observáveis, uma função de erro ou uma função distância.

Quando $J(\hat{\omega}) = \min_{\omega} J(\omega)$, onde $\hat{\omega}$ representa a estimativa de ω pelo critério *Least Square*. No caso específico da estimação do contorno da elipse, $\omega = \mathbf{a}$ representa os parâmetros implícitos da elipse parametrizada.

A estimação do contorno de uma elipse, a partir de um conjunto de pontos, pode ser resolvido utilizando o método LS. Dado o conjunto de n-pontos (x, y), representando as coordenadas a serem consideradas no plano XY, é possível estimar os parâmetros de uma elipse, definida pela Equação (3.2), onde, aplicando a Equação (3.3), seu contorno minimize a distância para cada ponto, como ilustrado na Figura 3.2.

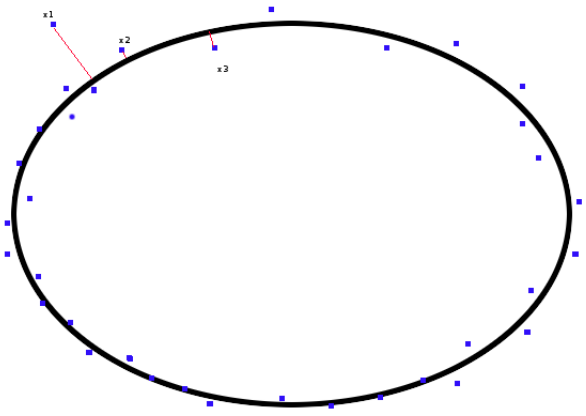

Figura 3.2: Definindo os parâmetros da elipse pelo métodos de mínimos quadrados.

[Boo79] apresenta uma solução não-trivial para a minimização da Equação (3.2) encontrando os autovalores λ definidos por

$$\mathbf{D^T\,D\,a} = \lambda\,\mathbf{C\,a} \quad , \tag{3.11}$$

onde $\mathbf{D} = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots \mathbf{x}_n]^T$ é a matriz com os n-pontos (x, y), $\mathbf{C}$ é a matriz de restrição

obtida da relação $4\,a\,c - b^2 = 1$, que na forma matricial é definida pela Equação (3.12), e $\mathbf{a}$ são os parâmetros da elipse a serem estimados

$$\mathbf{a^T\,C\,a} = 1 \quad , \tag{3.12}$$

onde

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad . \tag{3.13}$$

A partir das Equações (3.11) e (3.12) forma-se o sistema que se deseja resolver, dado por

$$\begin{cases} 2\,\mathbf{D}^T\mathbf{D}\,\mathbf{a} = 0 \\ \mathbf{a}^T\,\mathbf{C}\,\mathbf{a} = 1 \end{cases} \quad . \tag{3.14}$$

Por LS, minimizando $E = \| \mathbf{D}\,\mathbf{a} \|^2$, sujeito a restrição $\mathbf{a}^T\ \mathbf{C}\,\mathbf{a} = 1$, e fazendo $\mathbf{S} = \mathbf{D}^T\,\mathbf{D}$, o sistema definido em (3.14) é reescrito como

$$\begin{cases} \mathbf{S}\,\mathbf{a} = \lambda\,\mathbf{C}\,\mathbf{a} \\ \mathbf{a}^T\,\mathbf{C}\,\mathbf{a} = 1 \end{cases} \quad . \tag{3.15}$$

Se $(\lambda_i, \mathbf{u}_i)$ resolver $\mathbf{S}\,\mathbf{a} = \lambda\,\mathbf{C}\,\mathbf{a}$, então $(\lambda_i, \mu\,\mathbf{u}_i)$, para qualquer μ, resolve $\mathbf{a}^T\ \mathbf{C}\,\mathbf{a} = 1$, tal como apresentado por

$$\begin{aligned} (\mu_i\,\mathbf{u}_i)^T\,\mathbf{C}\,(\mu_i\,\mathbf{u}_i) &= 1 \\ \mu^2\,\mathbf{u}_i\,\mathbf{C}\,\mathbf{u}_i &= 1 \\ \mu_i = \sqrt{\frac{1}{\mathbf{u}^T\,\mathbf{C}\,\mathbf{u}_i}} &= \sqrt{\frac{1}{\mu^T\,\mathbf{S}\,\mu_i}} \quad . \end{aligned} \tag{3.16}$$

Finalizando, o valor estimado $\hat{a}_i = \mu_i\,\mathbf{u}_i$ soluciona o sistema de Equação (3.14).

A título de experimentação da técnica, são apresentados, abaixo, alguns resultados experimentais, na forma matricial, a partir de seis pontos x_i, $x_i \in \mathbb{R}^2$, definidos

como $x_1 = [82\ 53]^T$, $x_2 = [33\ 91]^T$, $x_3 = [47\ 148]^T$, $x_4 = [110\ 160]^T$, $x_5 = [163\ 121]^T$ e $x_6 = [150\ 58]^T$.

Design matrix:

$$\mathbf{D} = \begin{bmatrix} 6724 & 4346 & 2809 & 82 & 53 & 1 \\ 1089 & 3003 & 8281 & 33 & 91 & 1 \\ 2209 & 6956 & 21904 & 47 & 148 & 1 \\ 12100 & 17600 & 25600 & 110 & 160 & 1 \\ 26569 & 19723 & 14641 & 163 & 121 & 1 \\ 22500 & 8700 & 3364 & 150 & 58 & 1 \end{bmatrix}$$

Scatter matrix:

$$\mathbf{S} = \begin{bmatrix} 1.40985e+009 & 9.80589e+008 & 8.50738e+008 & 9.72788e+006 & 7.23825e+006 & 71191 \\ 9.80589e+008 & 8.50738e+008 & 9.58031e+008 & 7.23825e+006 & 7.24018e+006 & 60328 \\ 8.50738e+008 & 9.58031e+008 & 1.43729e+009 & 7.24018e+006 & 1.02069e+007 & 76599 \\ 9.72788e+006 & 7.23825e+006 & 7.24018e+006 & 71191 & 60328 & 585 \\ 7.23825e+006 & 7.24018e+006 & 1.02069e+007 & 60328 & 76599 & 631 \\ 71191 & 60328 & 76599 & 585 & 631 & 6 \end{bmatrix}$$

Cholesky matrix:

$$\mathbf{L} = \begin{bmatrix} 37548 & 0 & 0 & 0 & 0 & 0 \\ 26115.6 & 12988.9 & 0 & 0 & 0 & 0 \\ 22657.4 & 28202.5 & 11338 & 0 & 0 & 0 \\ 259.079 & 36.3579 & 30.4074 & 42.694 & 0 & 0 \\ 192.773 & 169.82 & 92.5934 & 32.6671 & 30.9484 & 0 \\ 1.896 & 0.832457 & 0.896388 & 0.849379 & 0.432551 & 0.0117594 \end{bmatrix}$$

L-1 matrix:

$$\mathbf{L}^{-1} = \begin{bmatrix} 2.66326e-005 & 0 & 0 & 0 & 0 & 0 \\ -5.35478e-005 & 7.69887e-005 & 1.5275e-012 & 0 & 0 & 0 \\ 7.99749e-005 & -0.000191504 & 8.8199e-005 & 0 & 0 & 0 \\ -0.000172973 & 7.08298e-005 & -6.2817e-005 & 0.0234225 & 0 & 0 \\ 7.1241e-005 & 7.57385e-005 & -0.000197574 & -0.0247233 & 0.0323118 & 0 \\ 0.00327366 & 0.00124582 & 0.00508155 & -0.782399 & -1.18854 & 85.0386 \end{bmatrix}$$

CC matrix:

$$\mathbf{C} = \begin{bmatrix} 0 & -8.136e-17 & -4.697e-09 & 3.345e-09 & 1.052e-08 & -2.706e-07 \\ -8.136e-17 & 5.927e-09 & -5.297e-09 & -1.274e-09 & -1.532e-08 & 6.401e-07 \\ -4.697e-09 & -5.297e-09 & 8.459e-09 & 2.699e-08 & 4.530e-09 & -1.628e-06 \\ 3.345e-09 & -1.274e-09 & 2.699e-08 & -3.844e-08 & -5.403e-08 & 2.257e-06 \\ 1.052e-08 & -1.532e-08 & 4.530e-09 & -5.403e-08 & 6.203e-08 & 6.639e-07 \\ -2.706e-07 & 6.401e-07 & -1.628e-06 & 2.257e-06 & 6.639e-07 & -6.498e-05 \end{bmatrix}$$

Autovalores:

$$\mathbf{e} = \begin{bmatrix} 6.88287e-008 \\ 4.9201e-008 \\ 3.99216e-008 \\ 1.22082e-008 \\ 1.12554e-009 \\ -6.51222e-005 \end{bmatrix}$$

Autovetores:

$$\mathbf{V} = \begin{bmatrix} 0 & 0 & 0.000255231 & -0.000354006 & 0.999948 & 0.0101976 \\ 0 & 0 & 0.999687 & 0.000867547 & 0 & -0.0249907 \\ 0 & 0 & 0 & 0.999398 & 0 & 0.0346939 \\ 0 & 0.999952 & 0.000245543 & -0.000340568 & -0.000100231 & 0.00981046 \\ 0.999991 & 4.08247e-005 & -0.000103946 & 0.000144174 & 4.24312e-005 & -0.0041531 \\ 0.00415732 & -0.00981987 & 0.025003 & -0.0346793 & -0.0102063 & 0.998977 \end{bmatrix}$$

Solução normalizada:

$$\mathbf{sol} = \begin{bmatrix} 9.97643e-07 & -2.41161e-06 & 3.32476e-07 & -3.94918e-07 & -7.94866e-08 & 3.84795e-005 \\ 9.51401e-07 & 6.88951e-07 & 1.27129e-06 & -2.75765e-06 & -1.49548e-07 & 1.45368e-005 \\ -2.07461e-06 & -1.32536e-06 & 1.49393e-06 & -1.03568e-06 & -6.09825e-07 & 5.97247e-005 \\ -0.000328931 & 0.000365706 & -0.000229911 & 0.000318887 & 9.38502e-05 & -0.0091859 \\ 0.000321814 & 0.000137244 & -0.000349446 & 0.000484682 & 0.000142645 & -0.0139618 \\ 0.00415673 & -0.00981849 & 0.0249995 & -0.0346744 & -0.0102049 & 0.998837 \end{bmatrix}$$

Capítulo 4

Definição da dinâmica dos elementos da cena

Este capítulo apresenta como é modelado o problema da estabilização de vídeo, pela ótica do *Particle Filter*, implementando um algoritmo para construir a estratégia de definição do *Importance Sampling* (IS), a partir da estimativa de movimento da face entre dois quadros consecutivos. Esta estimativa permite propor uma região interna ao quadro corrente (tempo k), que será utilizado para gerar as amostras das partículas do filtro que definirão o movimento global da câmera, quadro a quadro.

Inicialmente, é apresentada uma breve introdução conceitual da identificação de objetos e de estados em processamento de sinais em vídeo digital, seguido pela descrição do algoritmo de acompanhamento da face que implementa o *Importance Sampling*. Finalizando, o algoritmo de definição do movimento global da câmera é apresentado.

4.1 Introdução

Uma das dificuldades encontradas em algoritmos aplicados em computação visual decorre do fato de se tentar simular, a partir de uma separação hierárquica dos objetos da cena, a percepção visual humana [MK04]. Este, por sua natureza intrínsica, possui um contexto integrado do ambiente sem a necessidade de se definir uma hierarquia. Como ilustração, a percepção visual humana, quando identifica uma porta fechada, que inicia o movimento de abertura, tem uma interpretação perceptual contínua da porta da mesma maneira de quando ela estava fechada, independente de alteração da iluminação do ambiente ou

deslocamento do ponto de referência inicial do observador. Ao contrário dos algoritmos de visão computacional, que possuem uma hierarquia de objetos da cena que muda de estados no tempo discreto, tentar identificar a porta em movimento como um objeto que deforma sua superfície enquanto se move requer um algoritmo que estime o movimento da porta através da definição de um modelo que interprete a transição de estados no tempo discreto. Um algoritmo genérico que realize esta tarefa é muitas vezes difícil de se construir, devido a dimensão exigida pelo modelo, que ainda requer uma alta complexidade computacional.

Em Engenharia, é comum definir um modelo no Espaço de Estados que possibilite descrever e acompanhar, no caso de aplicações no campo da visão computacional, os objetos da cena de um vídeo digital. O foco aqui é a face do interlocutor, para a qual já foi proposto um modelo parametrizado no Capítulo 3. O que se propõe agora é apresentar o modelo do Espaço de Estados e aplicar a metodologia de *Particle Filter* para conseguir acompanhar, de forma robusta, a face do interlocutor e determinar qual a velocidade ou o deslocamento global da cena que ocorre inter-quadros, possibilitando compensá-lo e oferecendo uma percepção de um vídeo estabilizado.

4.2 Modelo do espaço de estados discreto

A partir do modelo definido para representar a face parametrizada através de uma elipse, como apresentado na Equação (3.5), é possível construir o espaço de estados representado pelo vetor de estados no tempo $\{\mathbf{x_k} : \Re^{\mathbf{4}},\ k \in \mathcal{N}\}$, tal que

$$\mathbf{x_k} = [\ x_c \quad y_c \quad b \quad \phi\ ]^T \quad , \tag{4.1}$$

onde (x_c, y_c) representa as coordenadas no quadro $\mathbf{z}_k$ onde encontra-se o centro da elipse, $\mathbf{b}$ é o eixo maior da elipse, e ϕ é sua orientação em radianos da elipse com o eixo de coordenadas horizontal. Neste modelo, é omitido o parâmetro que define o eixo menor da elipse, como descrito no Capítulo 3, conseguindo assim diminuir a dimensão do vetor de estados e, consequentemente, a complexidade computacional do algoritmo. A Figura 4.1 ilustra os parâmetros descritos acima.

Tomando como base o deslocamento da face ocorrido entre dois quadros consecutivos, decorrente, ou do movimento incondicional do interlocutor na cena, como ilustrado na Figura 4.2.a, ou de qualquer movimento aplicado à câmera, Figura 4.2.b, que ocasione uma deformação na cena, ou até mesmo ambos os eventos consecutivamente; é preciso

identificar o vetor de deslocamento da face, $\Delta\mathbf{x}_k$, que representa a diferença entre o vetor de estados $\mathbf{x}$ definido no instante $\mathbf{k}$, $\mathbf{x_k}$, e também no instante **k-1**, $\mathbf{x_{k-1}}$, como descrito na Equação (3.7), reescrita para melhor compreensão na Equação (4.2).

$$\Delta\mathbf{x_k} = [\ x_c(k) - x_c(k-1) \quad y_c(k) - y_c(k-1) \quad 0 \quad 0\ ]^T \quad . \tag{4.2}$$

Considerando na Equação (4.2), a hipótese de que os eixos e a orientação da elipse que representa a face não variam devido ao pequeno intervalo de tempo das amostras consecutivas, o problema de acompanhamento da face pode ser modelado como

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta\mathbf{x}_k + \upsilon_k \quad , \tag{4.3}$$

onde υ_k representa o ruído aditivo modelado como uma função Gaussiana de média zero e variância Σ_G, $\upsilon_k = \mathcal{N}(0, \Sigma_G)$, que pode ser interpretado como o erro de precisão de medida para determinar a posição exata da face no instante de tempo k.

Uma importante consideração final sobre o vetor de deslocamento $\Delta\mathbf{x}_k$, definido na Equação (4.2), é considerar seu comportamento tal como processo de Markov, não linear, de primeira ordem. Como um processo não linear, sua estimativa de estados através de um modelo puramente linear deve sofrer aproximações que acarretam num erro perceptível em relação ao estado real. Como processo de Markov de primeira ordem, o estado corrente do vetor $\mathbf{x}$, $\mathbf{x}_k$, depende apenas do seu estado anterior $\mathbf{x}_{k-1}$. Essas duas características tornam adequado o emprego da metodologia *Particle Filter* na solução de uma estimativa robusta e precisa do acompanhamento da face.

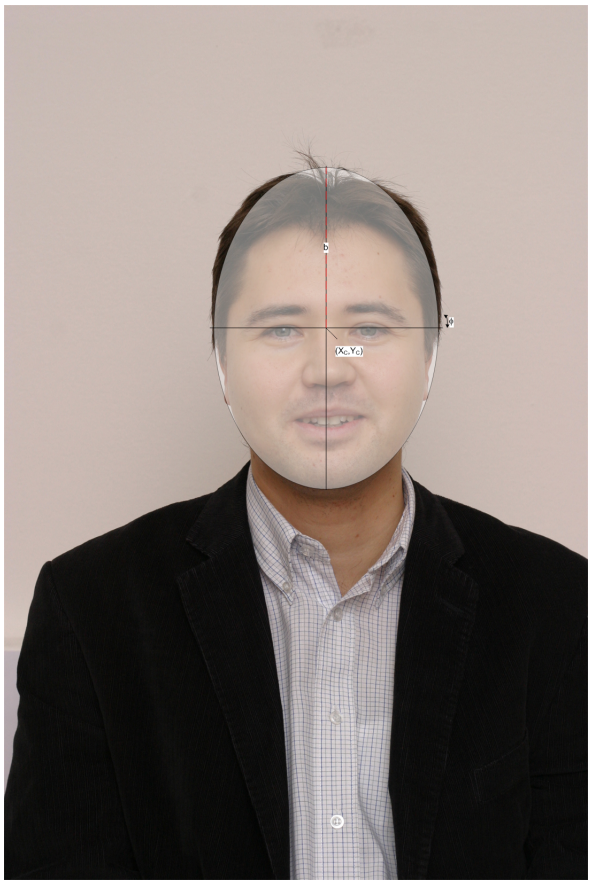

Figura 4.1: Representação do espaço de estados da face do interlocutor.

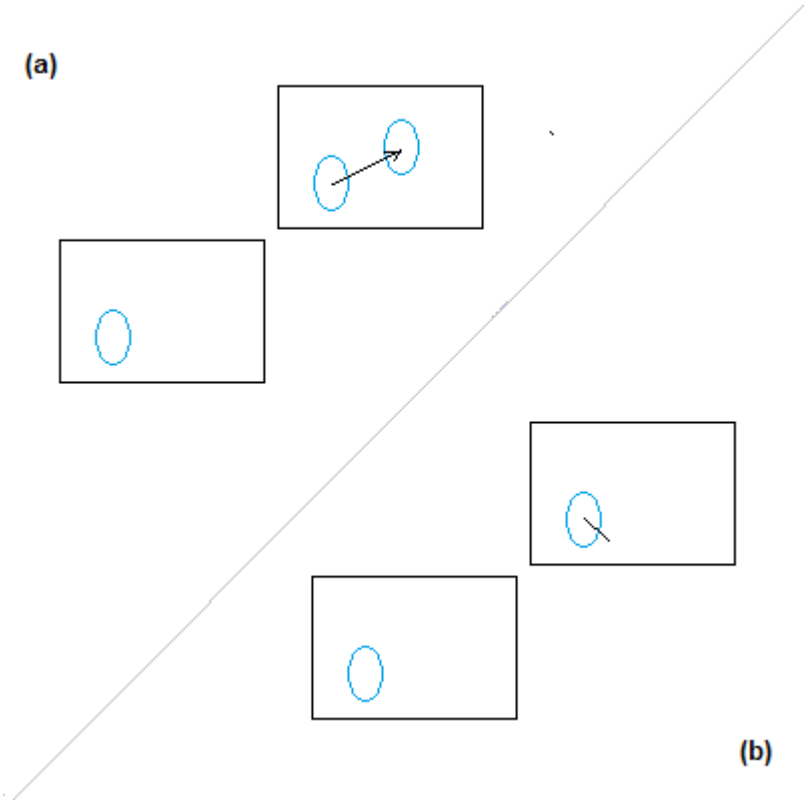

Figura 4.2: (a) Translação da face. (b) Translação e rotação da câmera, alterando a origem de referência enquanto a face permanece imóvel.

4.3 Bayesian importance sampling

O deslocamento da face, ou qualquer outro objeto, em um vídeo é facilmente acompanhado pela visão humana [MK04], que abstrai os detalhes da cena, calculando naturalmente o deslocamento *pixel* a *pixel* que define o movimento observado. Entretanto, este cálculo é extremamente difícil quando realizado por um computador, que necessita calcular numericamente o deslocamento da face, através de um algoritmo que será executado por um programa em memória. É necessário, então, representar o objeto de interesse através de parâmetros que possam ser calculados por um algoritmo computacional e que represente uma distribuição de probabilidade proporcional à encontrada no deslocamento real *pixel* a *pixel*. A cada novo quadro de vídeo, os objetos da cena estão deformando, rotacionando, transladando, alterando sua escala (*zoom*), em relação a um quadro inicial, dado como referência. Todas essas trasformações espaciais, aplicadas a cada quadro, podem ser descritas matematicamente por uma transformação *affine* [GW08].

Como visto no Capítulo 2, o que se deseja representar num problema de acompanhamento, através do emprego de uma técnica recursiva de filtragem Bayesiana, é a função de densidade *a posteriori* marginal $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. Com $\mathbf{z}_{1:k}$ representando o histórico de amostras, desde a primeira observação $\mathbf{z}_1$ até a observação corrente $\mathbf{z}_k$. Num espaço de estados Markoviano de primeira ordem, discreto no tempo, podemos esboçar a função de densidade *a posteriori* através de um algoritmo baseado em simulação Monte Carlo, onde são geradas amostras aleatórias de distribuição uniforme do vetor de deslocamento da face $\Delta\mathbf{x}_k$, Equação (4.2). Entretanto esse enfoque mostrou-se ineficiente, visto que a densidade *a posteriori* do sistema proposto é representada pela face que se deseja acompanhar. Obter amostras uniformemente distribuídas do vetor de deslocamento resulta em construir hipóteses de estado da face sem considerar a observação corrente do sistema obtida no quadro $\mathbf{z}_k$. Uma técnica que propõe um novo esquema de amostragem aleatória a partir de uma distribuição não-uniforme é conhecida pelo nome de *Importance Sampling* [AMGC02].

No escopo deste trabalho, definimos a *Importance Function* baseada na informação de movimento do objeto que se deseja acompanhar. Essa abordagem permite simplificar o algoritmo de busca e, consequentemente, o tempo de processamento total quadro a quadro. Definindo-se uma área de processamento no quadro $\mathbf{z}_k$ onde é mais provável encontrar a face possibilita-se o processamento do *Particle Filter* gerar amostragens aleatórias numa

região limitada da cena, ou seja, amostras aleatórias não uniformemente distribuídas em relação a todo o espaço definido pela região total do quadro $\mathbf{z}_k$, contribuindo para que obtenhamos amostras mais próximas da nova posição da face no quadro corrente, ou seja, amostras com peso significativo (*Importance Weight*).

A Figura 4.3 ilustra a utilidade de se definir uma função de importância (*importance function*) adequada. O pior caso ocorre quando a função de importância não é definida ou está definida como uma função de densidade uniforme como apresentada na Figura 4.3.a. O algoritmo de acompanhamento da face deverá gerar as amostras aleatórias uniformemente em todo o quadro. Por outro lado, quando a função de importância é devidamente definida por uma função de probabilidade conhecida, como, por exemplo, uma distribuição gaussiana, a geração das hipóteses de estado da face fica restrita à região definida pela função de importância $q(\mathbf{x}_k|\mathbf{z}_k)$, promovendo uma eficiência maior ao processamento da estimação pelo filtro, tal como pode ser visto na Figura 4.3.b.

Em relação aos elementos do modelo apresentado, deve-se observar que, quando um modelo baseado no espaço de estados é proposto, ele deve ser descrito a partir de, pelo menos, duas equações distintas, além de outras equações auxiliares. Como descrito na Seção 2.2, a primeira equação que define o espaço de estados é conhecida por, entre outros nomes: equação de transição ou equação de estados, e é representada pela Equação (2.13). Ela descreve como a grandeza física que queremos apresentar comporta-se no tempo. No caso específico deste trabalho estamos representando a equação de estados como a face do interlocutor, modelada parametricamente por uma elipse. A segunda equação necessária é conhecida pelo nome de equação de medida (*measurement equation*) ou de observação, definida na Equação (2.14). A equação de observação está associada a cada quadro amostrado $\mathbf{z}_k$, pertencente ao vídeo, e representa um vetor de amostras bidimensional, $\mathbf{z}_k \in \Re^2$, contendo $m \times n$ elementos, onde m e n são número inteiros que representam, respectivamente, a resolução horizontal e vertical, em *pixels*, de cada quadro de vídeo, nos três diferentes canais R, G e B. Adicionalmente às equações descritas, pode ser definido um modelo de restrição (*constraint*) à equação de estados. Normalmente, esta equação tem a finalidade de reduzir a dimensão (dimensionalidade) do problema original, através do conhecimento prévio (*a priori knowledge*) de como a equação de transição se comporta no tempo. Na modelagem do problema proposto aqui são aplicados duas restrições distintas. A primeira restrição refere-se à proporção entre o eixo maior da elipse,

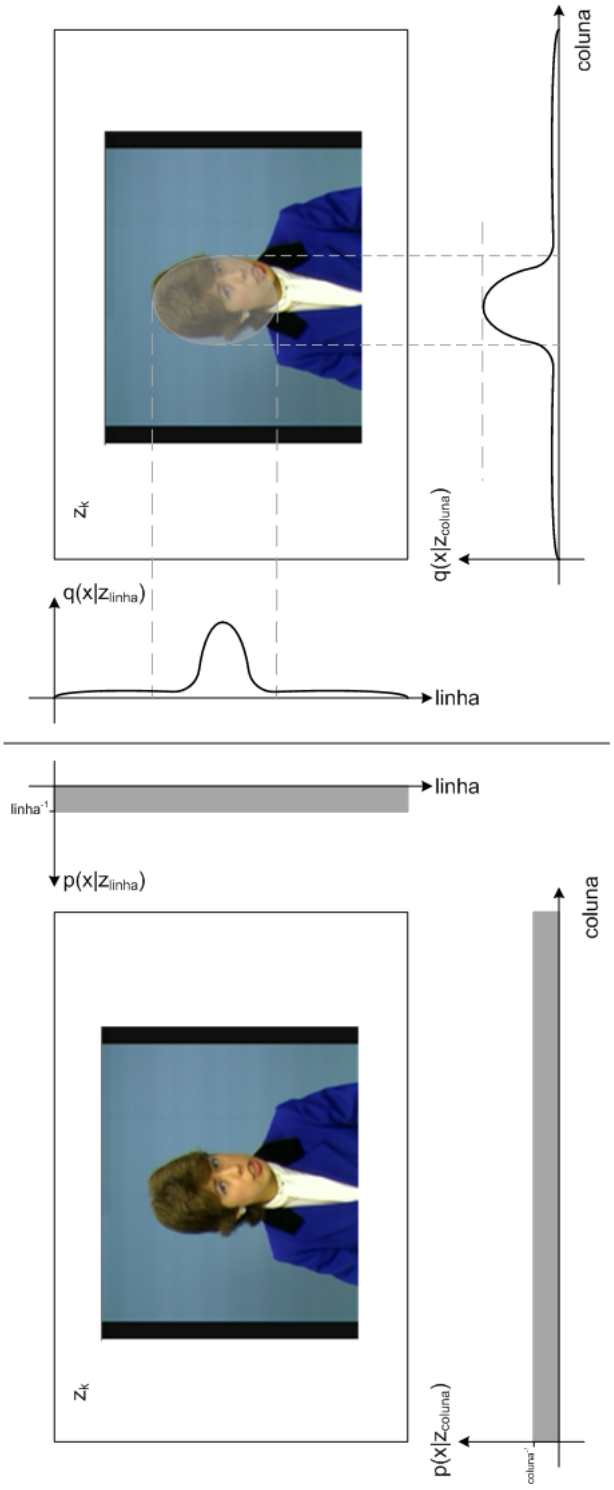

Figura 4.3: *Importance Function*: (a) O quadro da sequência de vídeo Claire sem a localização estimada da face obriga uma amostragem aleatória uniformemente em todo o espaço do quadro. Nesse caso, não há uma função de importância como pode ser visto nos gráficos da função de densidade horizontal e vertical. (b) O mesmo quadro apresentado em (a), porém indicando a região de maior probabilidade de localização da face, representada pela função de importance $q(x|z)$.

definido pelo parâmetro **b** da Equação (4.1), em relação ao seu eixo menor a, reduzindo a dimensão do vetor de estados $\mathbf{x}_k$ de cinco para quatro dimensões, $f_x : \Re^5 \rightarrow \Re^4$. A segunda restrição adotada é caracterizada pelo método utilizado para encontrar o vetor de deslocamento da face inter-quadros $\Delta\mathbf{x}_k$. O método será apresentado ainda neste capítulo e tem como finalidade reduzir a área, dentro de $\mathbf{z}_k$, da posição estimada aproximada da face, ou seja, a posição mais provável onde se encontra a elipse que representa a face, tendo como referência a posição em que estava definida a mesma no quadro anterior $\mathbf{z}_{k-1}$. Essas duas restrições definem o comportamento da função de importância.

Na Equação (4.3) foi definido que a nova posição da face $\mathbf{x}_k$ em relação à posição anterior $\mathbf{x}_{k-1}$ é representada pelo deslocamento $\Delta\mathbf{x}_k$ somado de um ruído de medida υ_k. A partir deste definição, é possível aplicar a Equação (2.65) e gerar as amostras aleatórias $\mathbf{X}_k^i$ que compõem as partículas previstas pelo *Particle Filter* seguindo o comportamento

$$\mathbf{X}_k^i = \mathbf{X}_{k-1}^i + \hat{\Delta\mathbf{x}}_k + \upsilon_k^i \quad , \tag{4.4}$$

tal como será abordado na Seção 4.4.4, onde, $\hat{\Delta\mathbf{x}}_k$ é a estimativa de deslocamento entre a posição da face no quadro $\mathbf{z}_{k-1}$ e $\mathbf{z}_k$, Equação (3.7); $i = 1, \ldots, N_p$ define o índice para cada um das N_p partículas $\mathbf{X}_k^i$ pertencente ao *Particle Filter*, como definido na Seção 4.4.4, e υ_k^i é o ruído de amostragem de média zero e variância Σ_G definido pela função gaussiana

$$\upsilon_{\mathbf{k}} = \mathcal{N}(0, \Sigma_G) \quad .$$

Com a explanação do procedimento de como são geradas as partículas $\mathbf{X}_k^i$, é possível representar a *Importance Sampling* como uma função seguindo uma distribuição conhecida. Nesse trabalho, optou-se por utilizar uma distribuição gaussiana onde a média da sua distribuição é definida pelo estado da i-ésima partícula $\mathbf{X}_{k-1}^i$ em $k-1$ adicionado ao vetor de deslocamento $\hat{\Delta\mathbf{x}}_k$ e onde sua variância Σ_G é a mesma definida para a componente de ruído υ_k. A *Importance Sampling* utilizada para construir as partículas do *Particle Filter* é dada por

$$q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k) = \sum_{r=1}^{N_r} \sum_{i=1}^{N_p} N_{x_k}(\mathbf{X}_{k-1}^{<i>} + \Delta\mathbf{x}_k^r, \Sigma_G) \quad , \tag{4.5}$$

onde $N_{x_k}(.)$ representa a função Gaussiana

$$N_x(\mu, \Sigma) \equiv \frac{1}{2\mu|\Sigma|} \exp(-(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)) \quad , \tag{4.6}$$

com $r = 1, \ldots, N_r$ representando os $N_r \geq 1$ possíveis deslocamentos da face dentro da região do quadro $\mathbf{z}_k$ e $i = 1, \ldots, N_p$ representa o índice das partículas definidos para o filtro, com N_p sendo o núemro total de partículas. O Apêndice B descreve a função Gaussiana apresentada na Equação (4.6) em detalhes.

Portanto, a proposta de construir uma amostragem seguindo o método *Importance Sampling* fica apresentado conforme a Equação (4.5). A Figura 4.4 ilustra os elementos constituintes do método *Particle Filter*. A elipse em vermelho apresenta a região do quadro onde se registra a maior probabilidade de ser localizado a face do interlocutor. As elipse em azul ilustram as diversas partículas geradas aleatoriamente, seguindo o esquema de amostragem Monte Carlo, definido pela Equação (4.5). Na Seção 4.4 será apresentado o algoritmo para a construção da *Importance Function* que será utilizado para gerar as partículas do *Particle Filter*.

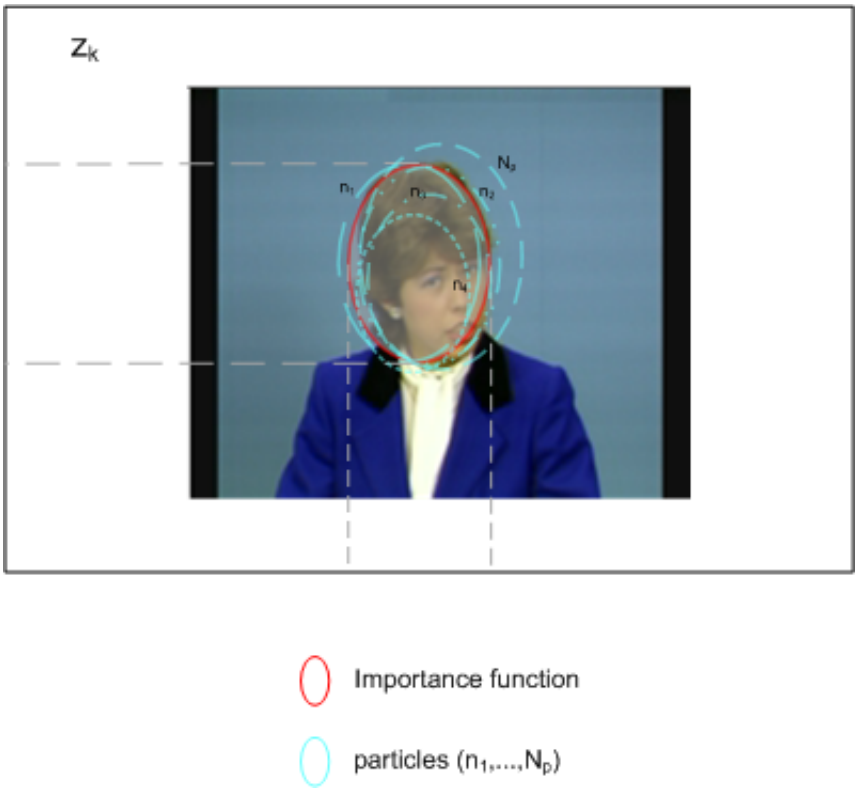

Figura 4.4: Ilustração dos elementos constituintes do *Particle Filter*. A elipse em vermelho representa a região na imagem definida como a região mais provável de localização da face definida pelo estágio de *Importance Function*. O conjunto de elipses de cor azul definem os diferentes *particles* amostrados aleatoriamente pelo algoritmo, com base na distribuição definida para o *Importance Sampling*.

4.4 Acompanhamento da face

4.4.1 Introdução

Para se determinar, em tempo real, com precisão e robustez, o deslocamento da face do interlocutor, é preciso calcular a *Importance Function* através da informação de movimento calculada entre dois quadros consecutivos. *Block Matching* é a técnica amplamente difundida em processamento de vídeo digital para a estimação de movimento entre quadros consecutivos [BS04]. A implementação clássica deste método compara dois quadros consecutivos $\{\mathbf{z}_{k-1}, \mathbf{z}_k\}$, pertencentes a uma mesma sequência de vídeo. O quadro $\mathbf{z}_{k-1}$ é dividido em blocos sequenciais $n \times n$ e cada um deles é comparado, por um critério de comparação pré-definido, com um subconjunto de blocos do quadro $\mathbf{z}_k$. O intuito é encontrar em $\mathbf{z}_k$ o bloco de maior similaridade e, assim, calcular o deslocamento no tempo de um grupo de *pixels*. De posse da informação de similaridade entre blocos de diferentes quadros, pode-se estimar, entre outros parâmetros, o deslocamento global inter-quadros. Nesta dissertação, será utilizada uma variação deste método para determinação da nova região no plano da imagem, onde é mais provável encontrar a face do interlocutor. O método foi batizado como "Blocos Particionados". Na técnica proposta, o quadro $\mathbf{z}_k$ é dividido em blocos $n \times n$ contíguos e apenas uma fração dos *pixels* contidos em cada bloco $n \times n$ é utilizado para determinar se o mesmo pertence à região onde encontra-se o objeto a ser localizado.

A motivação para o uso do método Blocos Particionados surge em estimar o deslocamento desejado de maneira eficiente e com o menor custo possível. Do mesmo modo que o *Block Matching*, cada quadro de vídeo é dividido em blocos. Entretanto, apenas os *pixels* pertencentes à borda de cada bloco são utilizados para o cálculo da localização da nova posição da face. Este grupo de *pixels* é então comparado com o histograma de cor da face de um modelo previamente construído durante a sua identificação, ou qualquer outro objeto a ser acompanhado, identificando se cada bloco da imagem no tempo k pertence ou não à região da face. Esta simplificação na busca pela região do quadro onde se localiza a face do interlocutor é adequada ao processamento em tempo-real, considerando ainda que a busca pela localização da face pode ser aplicada em toda a região do quadro $\mathbf{z}_k$, mesmo em sequências de vídeo de alta resolução, deixando para o processamento do *Particle Filter* a estimação da localização do objeto com a precisão de erro esperada.

A técnica empregada neste trabalho para calcular o histograma de cor é apresentada na Seção 4.4.2, enquanto que as Seções 4.4.3, 4.4.4 e 4.4.5, descrevem, respectivamente, o método Blocos Particionados, a implementação dos elementos do *Particle Filter* e a implementação da compensação do movimento indesejado que permite estabilizar a sequência do vídeo digital.

4.4.2 Histograma de cor

Cada quadro $\mathbf{z}_k$ da sequência de vídeo digital possui três canais de cor (RGB) [GW08], sendo 8 bits por *pixel* (bpp) em cada canal, com valores possíveis entre 0 e 255 ($R, G, B = 0, 1, 2, \cdots, 255$), que representam o modelo de cor utilizado para cada *pixel*. Os canais R, G e B são identificados como os canais de cor vermelho, verde e azul, respectivamente, e a resolução de cor para cada *pixel* é de $(2^8)^3$ possíveis combinações. A Figura 4.5 apresenta um quadro da sequência de vídeo "Claire" e ilustra como as componentes de cor R, G e B definem cada *pixel* pertencente ao quadro $\mathbf{z}_k$.

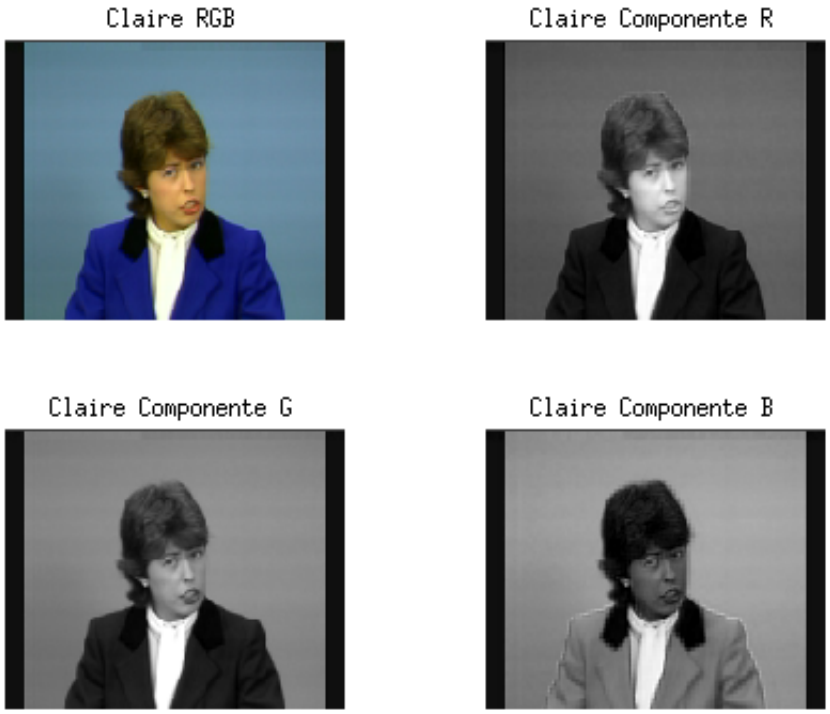

Figura 4.5: Quadros da sequência Claire: componentes RGB, componente R, componente G, componente B.

O método de cálculo do histograma de cor utilizado neste trabalho aplica uma transformação linear $(X, Y, Z) = \mathbf{T}\{(R, G, B)\}$ aos canais de cor R, G e B formando um novo espaço de cor X, Y e Z [Bir98]. A transformação é dada por

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \mathbf{B} - \mathbf{G} \\ \mathbf{G} - \mathbf{R} \\ \mathbf{R} + \mathbf{G} + \mathbf{B} \end{bmatrix} \quad . \tag{4.7}$$

Este método tem a preocupação de registrar, não somente a informação de crominância, como também a informação de luminância, aumentando a robustez na detecção das regiões no quadro $\mathbf{z}_k$ onde é encontrado a face, evitando, por exemplo, a similaridade entre o padrão de cor de cabelo castanho escuro e uma parede totalmente branca. A crominância representa a intensidade de tons de cinza que cada pixel possui, enquanto que a luminância, medida em lumens (lm), informa a quantidade de energia observada de uma fonte de luz e registra a informação de cor [GW08].

Manipulando-se (4.7), obtém-se a mesma transformação no formato $\mathbf{v} = \mathbf{A} \cdot \mathbf{x}$, mais conveniente para representar sistemas lineares, o qual é definido por

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} = \begin{bmatrix} 0 & -1 & 1 \\ -1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R} \\ \mathbf{G} \\ \mathbf{B} \end{bmatrix} \quad . \tag{4.8}$$

O modelo em (4.8) gera um novo conjunto de valores possíveis para cada canal, descrito por

$$\begin{aligned} X &= G - R, \quad X = -255, \ldots, 255 \\ Y &= B - G, \quad Y = -255, \ldots, 255 \\ Z &= R + G + B, \quad Z = 0, \ldots, 765 \ . \end{aligned} \tag{4.9}$$

No intuito de construir o histograma de cor, tal como apresentado por [SB91], é necessário definir os *bins* utilizados para cada canal X, Y e Z [Bir98]. Cada *bin* indica o número de níveis de quantização definido nas dimensões do histograma de cor. Todos os *bins* terão um espaçamento uniforme em cada canal X, Y e Z. Os canais X e Y contêm a informação de crominância e serão amostrados no histograma com uma resolução de oito *bins*, já o canal Z, responsável por armazenar a informação de luminância, será amostrado com resolução de quatro *bins*.

Dessa forma são definidos os *bins* x, y e z que mapeiam os valores atribuídos para cada canal X, Y e Z respectivamente. A Tabela 4.1 apresenta a faixa de valores definidos para cada um dos *bins* existente no histograma.

Tabela 4.1: Intervalos definidos para cada *bin* do histograma de cor.

canal	*bins*	1	2	3	4	5	6	7	8
X	x	-255..-192	-191..-128	-127..-64	-63..0	1..64	65..128	129..192	193..255
Y	y	-255..-192	-191..-128	-127..-64	-63..0	1..64	65..128	129..192	193..255
Z	z	0..190	191..382	383..574	575..765	—	—	—	—

A Figura 4.6 ilustra o conjunto de *bins* definido para cada canal e a distribuição deles no eixo de coordenadas $\Re^3$.

O histograma calcula a ocorrência em cada conjunto de *bins* (x,y,z) num espaço tridimensional. Cada elemento do quadro representado pela matriz $[X\ Y\ Z]^T$ pertence a um *bin*.

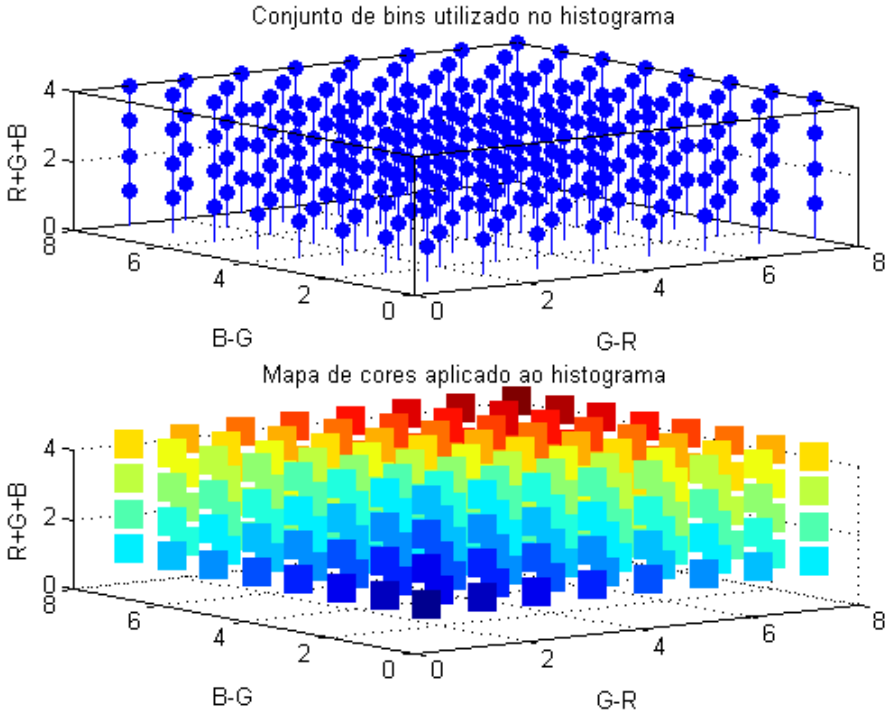

Figura 4.6: Mapa de *bins* definido em cada canal $X = G - R$, $Y = B - G$ e $Z = R + G + B$, tendo cada canal 8, 8 e 4 *bins*, respectivamente.

Para ilustrar a distribuição espacial do histograma de cor apresentado, três quadros utilizados nesta dissertação são analisados. Em primeiro lugar, a Figura 4.7 apresenta o histograma obtido da imagem “Lena”, ilustrada na Figura D.1. A seguir, a Figura 4.8 apresenta o histograma de um quadro da sequência de vídeo “Claire”, ilustrado na Figura D.3. Por fim, a Figura 4.9 apresenta o histograma da imagem “Alcatraz”, ilustrada na Figura D.2.

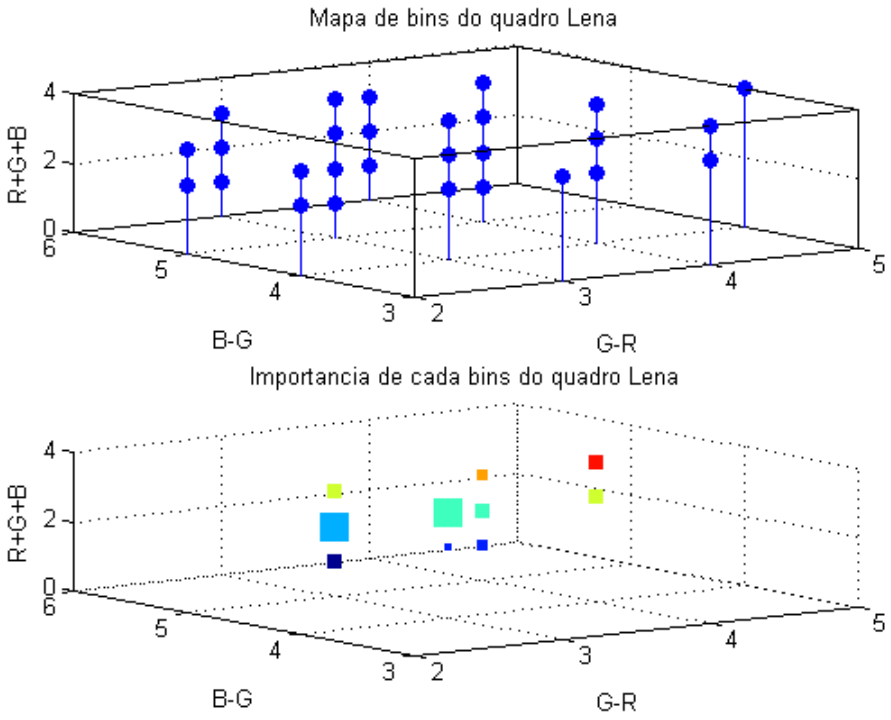

Figura 4.7: Histograma completo da imagem Lena.

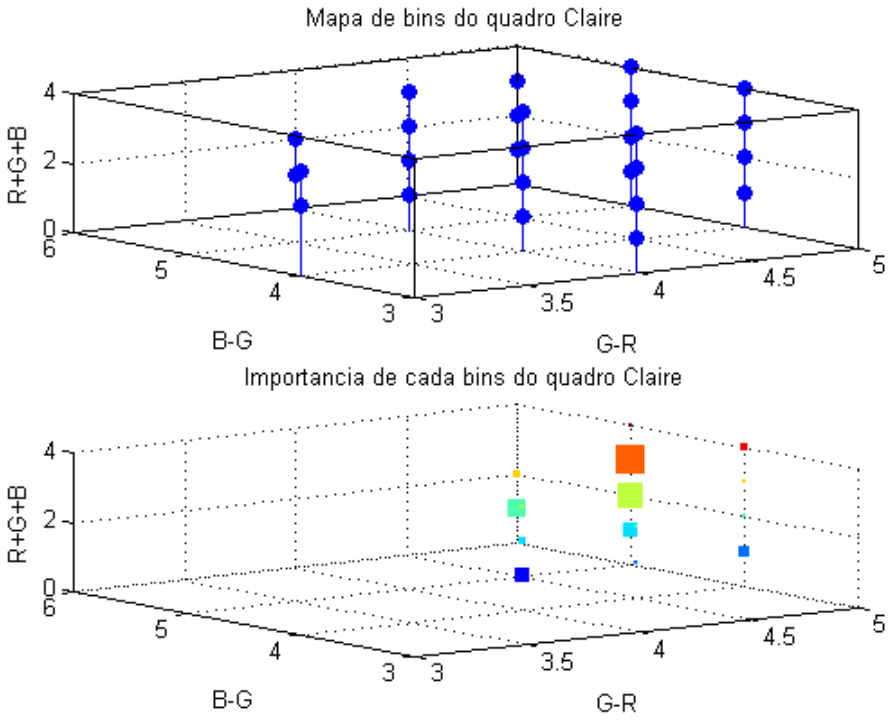

Figura 4.8: Histograma completo de um quadro do vídeo Claire.

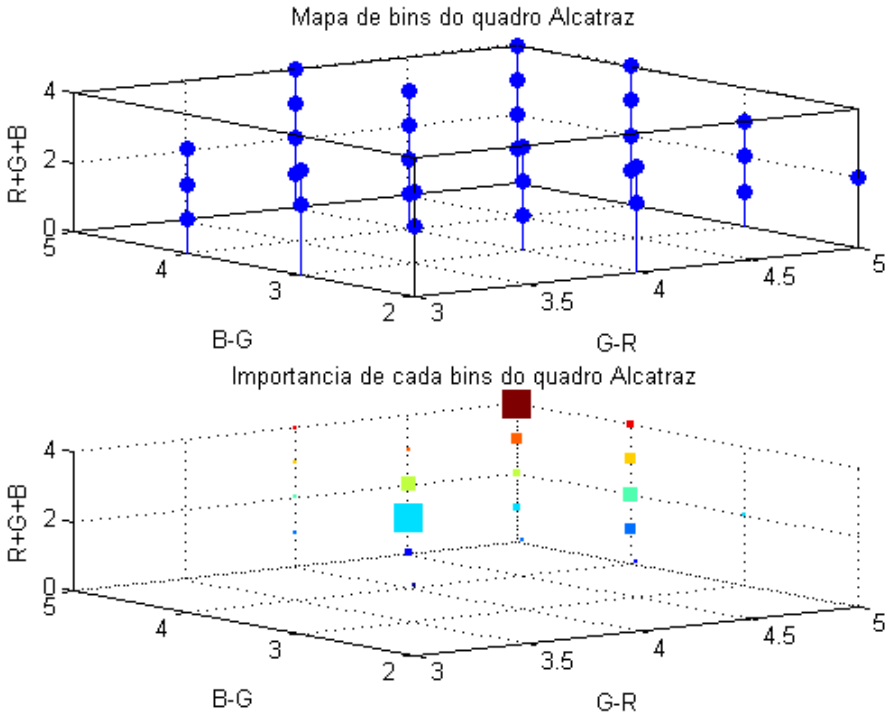

Figura 4.9: Histograma completo da imagem Alcatraz.

4.4.3 *Importance function*: método Blocos Particionados

A robustez e a precisão do filtro são diretamente proporcionais à escolha da *Importance Function*. Quanto mais próximo do estado real do objeto de acompanhamento for a *importance function* proposta, menor será o número de amostras N_P utilizadas pelo *particle filter* para representar a densidade de probabilidade *a posteriori* do sistema dinâmico, com a estimativa de erro (ϵ_k) esperada.

Esta dissertação propõe a técnica Blocos Particionados para a definição da *importance density* $q(\cdot)$, descrita na Equação (2.65), baseado no histograma de cor associado com o particionamento do quadro $\mathbf{z}_k$ em blocos de tamanho 32×32 *pixels*, procurando atender a três requisitos básicos definidos no escopo deste trabalho: (i) ser apropriado para qualquer resolução do quadro de vídeo, atendendo inclusive a quadros em alta definição, sem aumentar significamente a complexidade do processamento numérico ; (ii) permitir que a técnica descrita seja aplicada em toda a extensão do quadro de vídeo e (iii) ser utilizada em aplicações em tempo-real.

A implementação do método inicia com a identificação da região onde está localizada a face do interlocutor num quadro de referência, ou quadro modelo, $\mathbf{z}_0$. A representação da face modelo, tal como descrita na Equação (3.5), é identificada pelo vetor de estados $\mathbf{x}_0$. Como, neste instante de tempo, ainda não foi iniciado o processamento da filtragem digital para o acompanhamento da face é atribuído o índice $k = 0$.

Dando continuidade ao processamento, no instante de tempo $k > 0$, o quadro $\mathbf{z}_k$ da sequência de vídeo digital possui três canais de cor RGB [GW08], de 8 *bits* cada, que representam o modelo de cor utilizado para cada *pixel*. Os canais R, G e B são identificados como os canais de cor vermelho, verde e azul, respectivamente, e a resolução de cor para cada *pixel* é de $(2^8)^3$ possíveis combinações, $R, G, B = 0, 1, \ldots, 255$. Independentemente da resolução do vídeo de entrada, cada quadro é dividido em diferentes blocos $b^k_{i,j}$, $i, j \in \mathbb{Z}$, de tamanho 32×32 *pixels*, possuindo, cada um, os três canais de cor previstos.

O interesse da técnica de particionamento de blocos está em identificar quais blocos $b^k_{i,j}$ possuem o segmento do espaço $\mathcal{S}$ que corresponde ao objeto de acompanhamento definido pelo vetor de estado $\mathbf{x}_0$, ou seja, localizar os blocos que atendem à condição $b^k_{i,j} \subset \mathbf{x}_0 \subset \mathcal{S}$.

É assumido que, no instante $k = 0$, foi previamente calculado o histograma acumulativo da face do interlocutor $H_M(x, y, z; \mathbf{x}_0)$ [SB91], num espaço de cor tridimensional

representado pelos eixos $x = G - R$, $y = B - G$ e $z = R + G + B$, como a transformação apresentada na Equação (4.7). O histograma de cor tridimensional é definido por

$$H_M(x, y, z) = \sum_{i=1}^{x} \sum_{j=1}^{y} \sum_{k=1}^{z} H(x, y, z) \quad , \tag{4.10}$$

onde $H(x, y, z)$ é o histograma não acumulativo dos *pixels* de uma região definida por $\mathbf{x}_0$, [SB91]. Este histograma será o modelo utilizado para comparar os diversos blocos contidos no conjunto de quadros de vídeo amostrados nos intervalos de tempo $k \geq 1$, $\{\mathbf{z}_k | k \geq 1\}$, e permitir a definição da região que possui a maior probabilidade de ser localizada a face.

A Figura 4.10 ilustra a representação da região onde está posicionada a face inicial $\mathbf{x}_0$ do interlocutor através de uma imagem binária, de mesma resolução que o vídeo original. A imagem binária é descrita com uma imagem possuindo apenas um canal de informação de dados, com os *pixels* podendo ter valores 0 ou 1. Os *pixels* que representam o interior da face, modelada pela elipse parametrizada $\mathbf{x}_0$, recebem o valor 1, enquanto que os *pixels* externos à face recebem valor 0, formando uma máscara a ser aplicada no quadro $\mathbf{z}_0$, para o cálculo do histograma de cor. As cores negra e branca foram utilizados para melhor visualização dos *pixels* de valor 0 e 1, respectivamente.

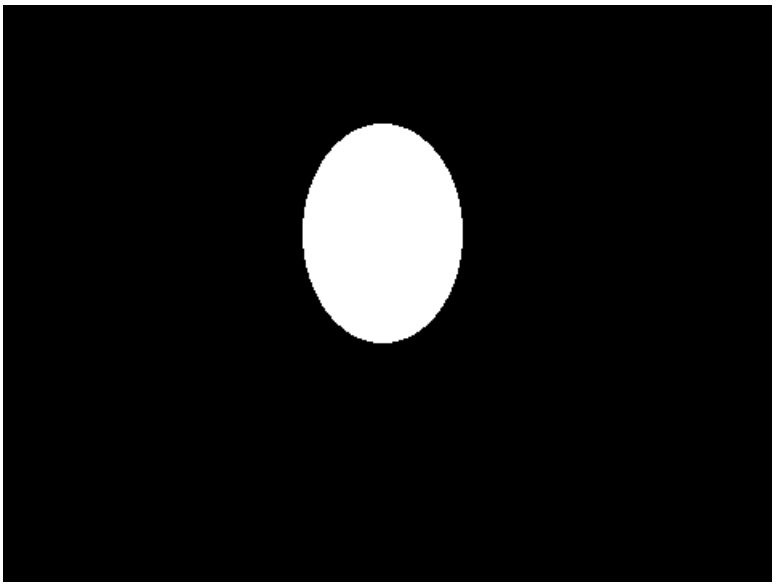

Figura 4.10: Máscara da face representada por uma elipse parametrizada.

Durante o processamento da filtragem digital, cada quadro de vídeo amostrado $\mathbf{z}_k$, $k > 0$, é dividido em blocos $b_{i,j}^k$, de tamanho 32×32 *pixels*, sendo i, j os índices de cada bloco interno ao quadro $\mathbf{z}_k$, $\{i, j \in N | i = n\frac{N_H}{N-1}, j = n\frac{N_V}{N-1}, N = 32, n = 0 : N - 1\}$,

tendo N_V e N_H as resoluções vertical e horizontal da sequência de vídeo, respectivamente.

A Figura 4.11 ilustra o particionamento em blocos sugerido neste trabalho. O quadro à esquerda mostra o quadro $\mathbf{z}_k$ completo, enquanto que o quadro à direita mostra quais os *pixels* em cada bloco serão utilizados no cálculo do histograma, como proposto nesta seção.

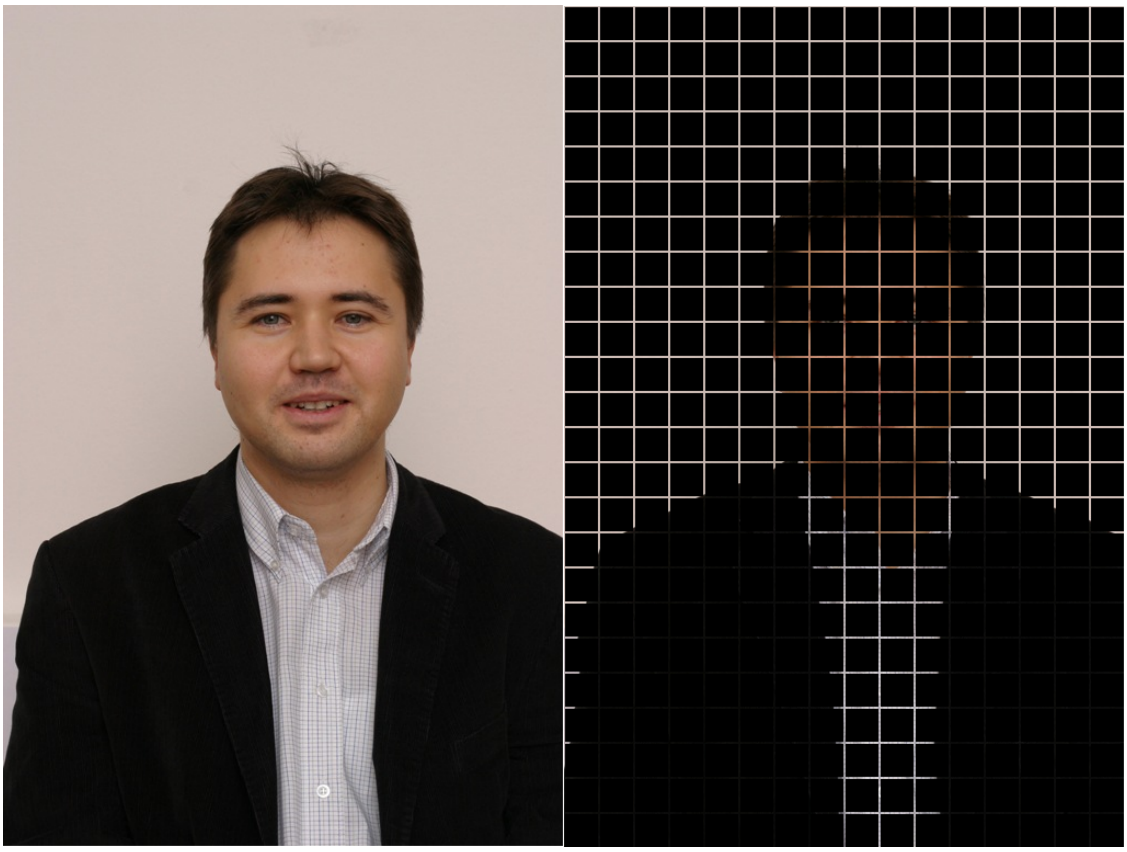

Figura 4.11: Particionamento de um quadro em blocos 32×32 *pixels*, apresentando apenas os pixels localizados na borda de cada bloco $b_{i,j}^k$.

Cada bloco é interpretado como um conjunto de *pixels* que representam um segmento da imagem original e que contém um subconjunto de informações de cor e gradiente pertencentes ao espaço de dados total do quadro $\mathbf{z}_k$.

O algoritmo Blocos Particionados calcula, para todos os blocos pertencentes ao quadro $\mathbf{z}_k$, o histograma apenas dos *pixels* pertencentes às suas respectivas bordas $H_{borda}(b_{i,j}^k)$, ao invés de todos os *pixels* interno ao bloco. A Figura 4.12 ilustra como os *pixels* de um bloco são selecionados para o cálculo do histograma de cor. Essa abordagem diminuiu significativamente o número de operações aritméticas necessárias para definir inicialmente as regiões do quadro de vídeo onde se encontram a face do interlocutor. Em seguida, cada

histograma da borda $H_{borda}(b_{i,j}^k)$ é comparado com o histograma do modelo da face $H_M(\mathbf{x})$ e somente aqueles histogramas das bordas que estão contidos no histograma do modelo, definido por

$$\begin{cases} H_M(\mathbf{x}) \supset H_{borda}(b_{i,j}^k) & , \text{ bloco válido} \\ H_M(\mathbf{x}) \not\supseteq H_{borda}(b_{i,j}^k) & , \text{ bloco inválido} \end{cases} , \tag{4.11}$$

serão considerados como aptos para a seleção da região da face.

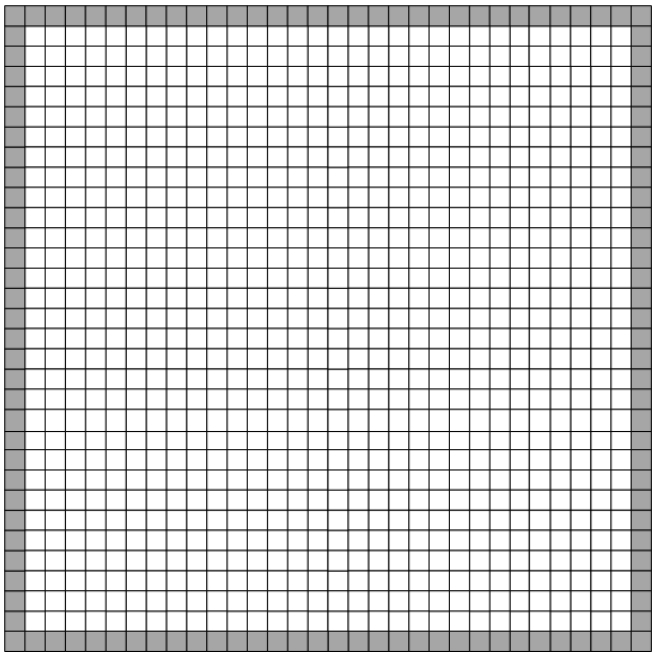

Figura 4.12: *Pixels* em cinza são selecionados em um bloco de dimensão 32×32 *pixels* para construir os histogramas $H_{borda}(b_{i,j}^k$ utilizados pelo método Blocos Particionados.

Para comparação dos histogramas de cor, foi utilizado o mesmo critério definido por [BS04]. A distância Bhattacharyya (d_B) [Cha] indica qual a similaridade que o bloco possui com a face do interlocutor e, portanto, torna-se um critério para a definição dos blocos selecionados. Ela é definida como

$$d_B(H_M(\mathbf{x}), H_{borda}(b_{i,j}^k)) = \sqrt{1 - \sum_i \frac{\sqrt{H_M^i(\mathbf{x}) H_{borda}^i(b_{i,j}^k)}}{\sqrt{\sum_i H_M^i(\mathbf{x}) \sum_i H_{borda}^i(b_{i,j}^k)}}} . \tag{4.12}$$

Assim, a Equação (4.11) pode ser reescrita, sem perda de generalidade, como

$$\begin{cases} d_B(H_M(\mathbf{x}), H_{borda}(b_{i,j}^k)) \geq T_{semelhanca} & , \text{ bloco válido} \\ d_B(H_M(\mathbf{x}), H_{borda}(b_{i,j}^k)) < T_{semelhanca} & , \text{ bloco inválido} \end{cases} . \tag{4.13}$$

Quando $d_B(H_M(\mathbf{x}), H_{borda}(b_{i,j}^k)) = 1$, existe uma semelhança perfeita entre o bloco $b_{i,j}^k$ e o região da face $\mathbf{x}_0$, indicando que os histogramas normalizados são idênticos. Quanto mais próximo de zero esta relação, menor será a semelhança entre os histogramas. Desse modo, um valor de *threshold* $T_{semelhanca}$, na faixa de $0 \leq T_{semelhanca} \leq 1$, é definido para identificar quais blocos da imagem possuem maior semelhança com a face e, portanto, serão considerados elegíveis para calcular o histograma completo $H(b_{i,j}^k)$, contendo todos os seus *pixels*.

De posse do conjunto de blocos $b_{i,j}^k$ localizados no quadro $\mathbf{z}_k$ no tempo k que pertence a região selecionada da face no quadro modelo em $k-1$, a próxima etapa é identificar o deslocamento

$$\Delta\mathbf{X}_k = \begin{bmatrix} \Delta x_c \\ \Delta y_c \\ 0 \\ 0 \end{bmatrix} \tag{4.14}$$

ocorrido pela face entre dois quadros consecutivos $\mathbf{z}_{k-1}$ e $\mathbf{z}_k$.

Ao considerar um subconjunto dos pixels contidos num bloco, o método Blocos Particionados caracteriza-se como uma técnica de subamostragem. Este tipo de abordagem constrói uma aproximação de cada quadro $\mathbf{z}_k$ que deve ser validado para assim calcular o deslocamento inter-quadros da face, objeto de acompanhamento deste trabalho.

A validação é realizada utilizando a técnica de *Block Matching* [Tek95], onde cada bloco válido $b_{i,j}^k$ é localizado no quadro anterior $\mathbf{z}_{k-1}$, utilizando somente a região $\mathbf{z}_{k-1} \supset \mathcal{S} \supset \mathbf{x}_{k-1}$ que se encontra a face.

O critério de comparação *Minimum Mean Absolute Difference* (MAD) foi empregado devido a sua popularidade e baixo custo em relação a outros critérios [Tek95], tal como o *Minimum Mean Square Error* (MSE). Assim, dado o bloco $b_{i,j}^k$ definido em k e a região $\mathcal{S}$ descrita pela localização da face $\mathbf{X}_{k-1}$ onde localiza-se a face no quadro $\mathbf{z}_{k-1}$ em $k-1$ e utilizando o critério

$$MAD_{b_{i,j}^k, b_{m,n}^{k-1}} = \frac{1}{N^2} \sum_{b_{i,j}^k, b_{m,n}^{k-1}} |b_{i,j}^k - b_{m,n}^{k-1}| \tag{4.15}$$

onde $b_{i,j}^k$ representa cada bloco pertencente ao conjunto de blocos válidos encontrados pelo método Blocos Particionados e $b_{m,n}^{k-1}$ representando um bloco localizado na região $\mathcal{S}$ em que localiza-se a face no instante de tempo $k-1$. Ambos blocos comparados possuem

a mesma dimensão e os quadros utilizados na comparação são formados pelo canal de luminância calculado a partir do quadro original RGB.

Utilizando o critério de busca exaustiva, o deslocamento da face é determinado como

$$\begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} = \arg \min_{b_{i,j}^{k}, b_{m,n}^{k-1}} MAD(b_{i,j}^{k}, b_{m,n}^{k-1}) \quad . \tag{4.16}$$

Assim, utilizando-se a Equação (4.16) é possível encontrar o vetor de deslocamento definido na Equação (4.2).

As Figuras 4.13 a 4.22 apresentam um exemplo ilustrando a comparação entre as distâncias de Bhattacharyya de um bloco posicionado no interior da região da face e outro fora desta região, a partir de um quadro de vídeo retirado da sequência "Claire", ilustrado pela Figura 4.5. Por fim, a Tabela 4.2 descreve todos os passos necessários para implementar o algoritmo dos "Blocos Particionados".

A Figura 4.13 apresenta o histograma em duas dimensões de cada componente de cor (RGB) da Figura 4.5. Nela, observa-se a distribuição do nível de intensidade de cada canal, tornando possível avaliar a importância de cada cor no cálculo do histograma através da utilização do modelo de cor proposto nesta dissertação, Equação (4.9).

A Figura 4.14 apresenta um quadro da sequência "Claire" dividido em blocos de tamanhos iguais, com exceção das extremidades, quando as dimensões do quadro não são múltiplos do tamanho do bloco, como definido anteriormente.

Dois blocos de mesma dimensão são selecionados na Figura 4.15. O primeiro bloco, identificado como b_1, foi selecionado de modo que não contenha qualquer informação referente à face, ou seja, o histograma deste bloco não está contido dentro do histograma definido como modelo, $H_{b_1}(\mathbf{x}) \nsubseteq H_M(\mathbf{x})$, ou ainda, $H_{b_1}(\mathbf{x}) \cap H_M(\mathbf{x}) = \mathbf{0}$. Por outro lado, o segundo bloco, identificado como b_2, foi especialmente selecionado para conter a maior parte da região da face possível, de tal forma que $H_{b_2}(\mathbf{x}) \subseteq H_M(\mathbf{x})$, ou ainda, $H_{b_2}(\mathbf{x}) \cap H_M(\mathbf{x}) > \mathbf{0}$.

Ao tentar analisar a afirmação de que os blocos selecionados na Figura 4.15 estão ou não na região da face, utilizando como critério os histogramas distintos para cada uma das componentes de cor R, G e B que formam cada quadro, assim como é apresentado nas Figuras 4.16 e 4.17, foi observado a sobreposição dos histogramas calculados para os canais R e G. Isso se deve ao fato de a luminância não ser considerada durante o cálculo de

cada histograma, tornando-o pouco robusto para o tipo de análise que se deseja realizar.

No caso do histograma sugerido na Equação (4.10), é possível classificar os blocos como parte integrante da face através da comparação dos histogramas de cada bloco selecionado com o modelo proposto. Na Figura 4.18, observa-se o histograma em três dimensões, calculado para todo o quadro do vídeo "Claire". A parte deste histograma correspondente ao bloco b_2 encontra-se presente na Figura 4.19, enquanto que o bloco b_2, referente a região externa à face, é apresentado na Figura 4.20. Assim, é possível utilizar o critério da distância Bhattacharyya como métrica de comparação entre os diferentes histogramas e identificar quais blocos pertencentem ou não à região da face definida no modelo.

As Figuras 4.21 e 4.22 apresentam o histograma somente das bordas dos blocos b_1 e b_2. Observe que o conjunto de *pixels* contidos nestas bordas são um subconjunto do conjunto total de *pixels* de cada bloco. Assim, pode-se afirmar que se o histograma das bordas de cada bloco sobrepuser o histograma da face selecionada pelo modelo, $H_{borda}(b^k_{i,j}) \cap H_M(\mathbf{x}) > \mathbf{0}$, então o bloco está parcialmente ou totalmente contido na região da face selecionada, tornando-o elegível a ser considerado no cálculo do *Importance Sampling*.

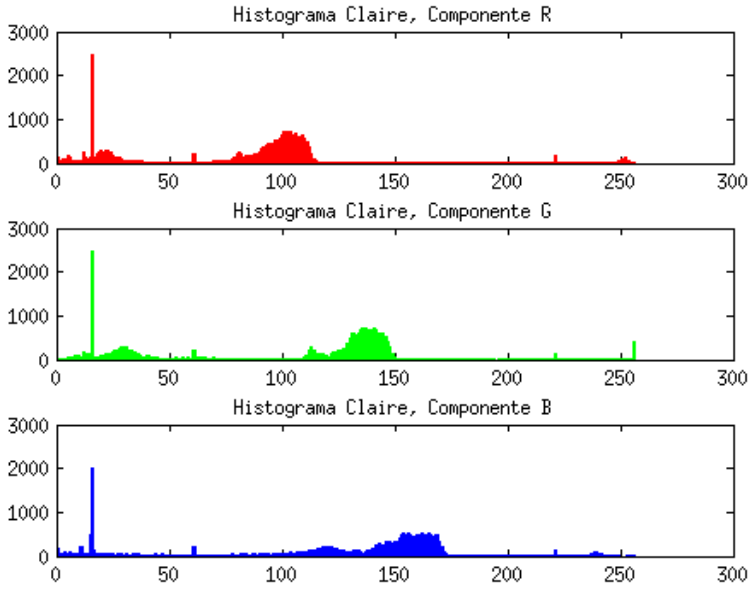

Figura 4.13: Histograma 2D de cada componente de cor (R, G, B), do quadro selecionado do vídeo Claire.

Figura 4.14: Quadro do vídeo Claire divido em blocos de 32×32 *pixels*.

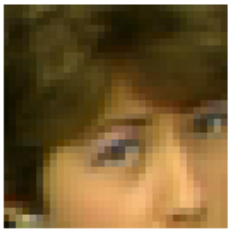

Figura 4.15: Blocos pertencentes à imagem Claire utilizados como referência. O primeiro bloco (topo) foi escolhido entre os blocos que não contêm qualquer informação da face. O segundo bloco (fundo) contém a região da face selecionada.

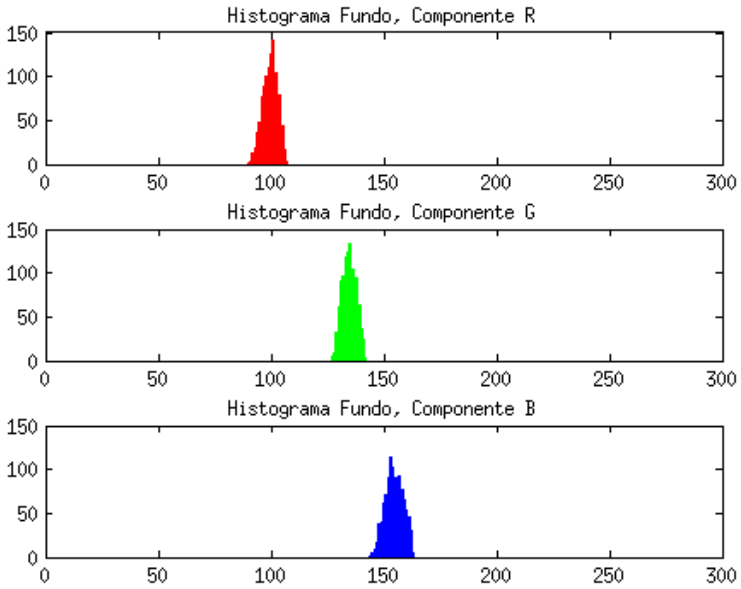

Figura 4.16: Histograma 2D do bloco externo à face pertencente ao quadro Claire.

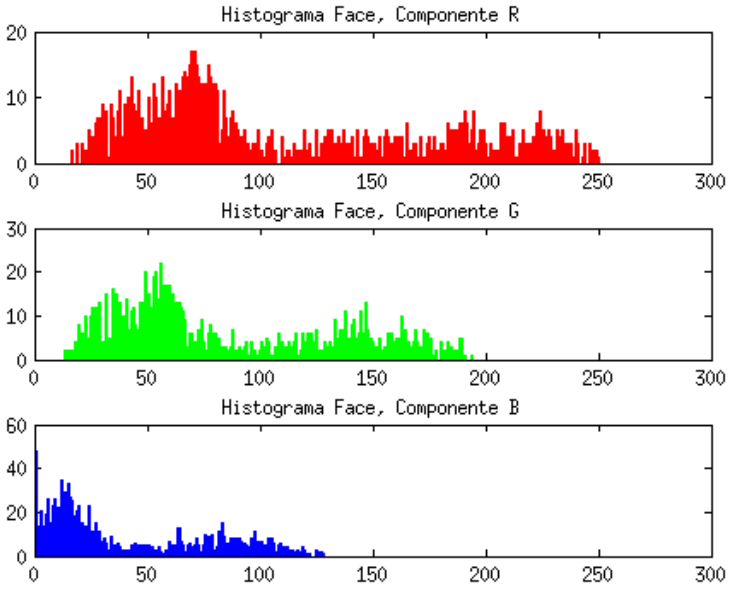

Figura 4.17: Histograma 2D do bloco interno à face pertencente ao quadro Claire.

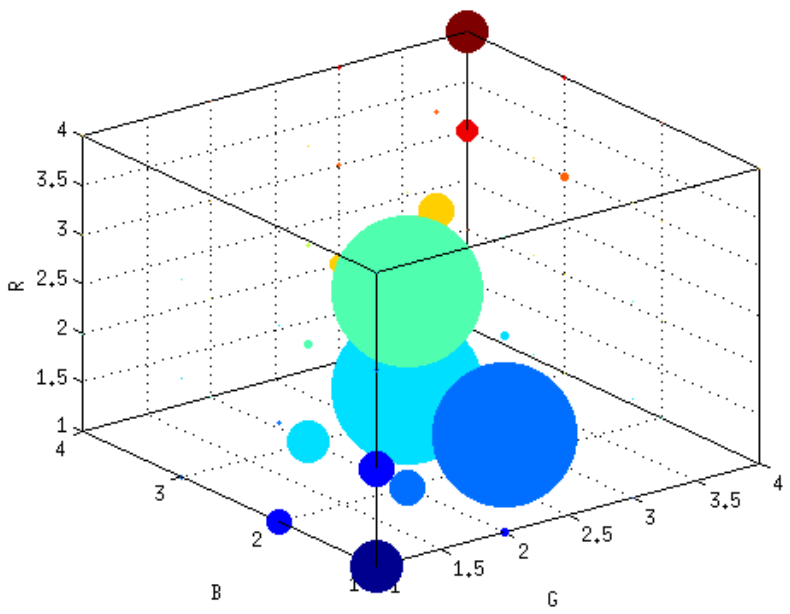

Figura 4.18: Histograma 3D do quadro Claire.

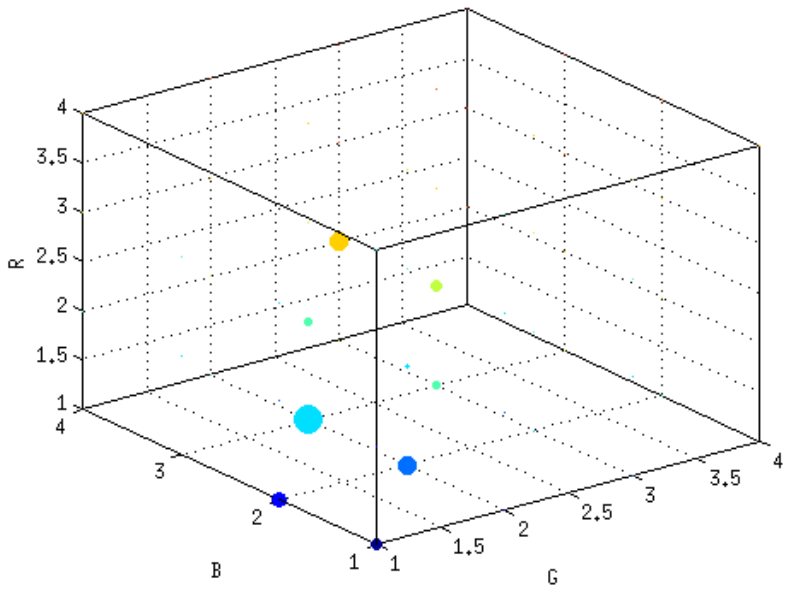

Figura 4.19: Histograma 3D de um bloco de 32×32 *pixels*, pertencente à face do quadro Claire.

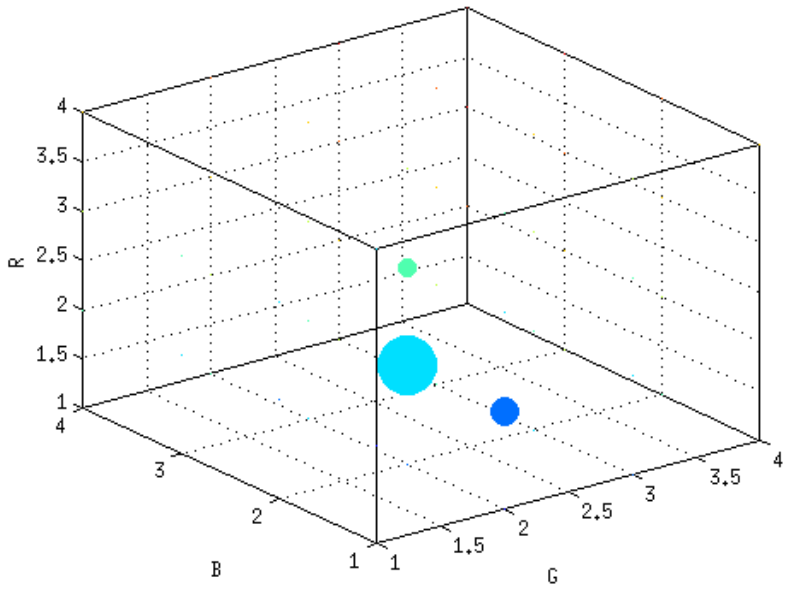

Figura 4.20: Histograma 3D de um bloco de 32 × 32 *pixels*, externo à face do quadro Claire.

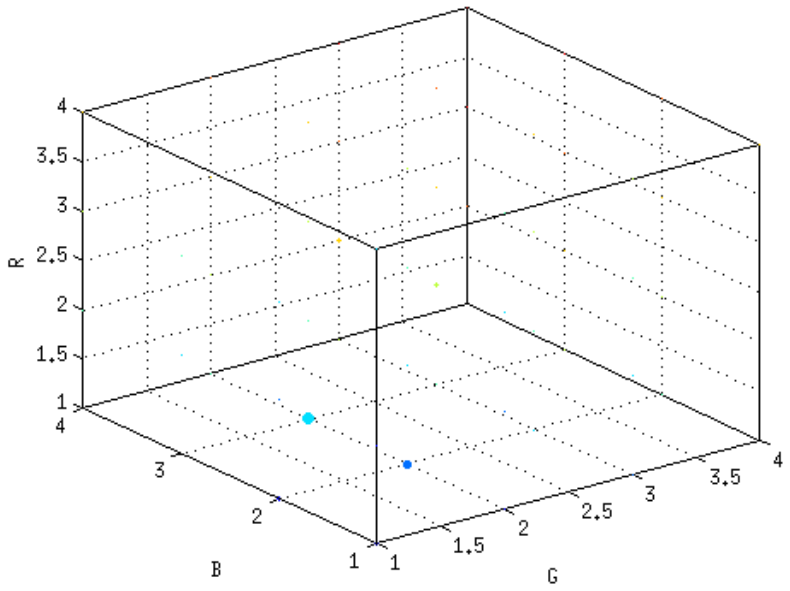

Figura 4.21: Histograma 3D das bordas do bloco de 32 × 32 *pixels*, interno a face do quadro Claire.

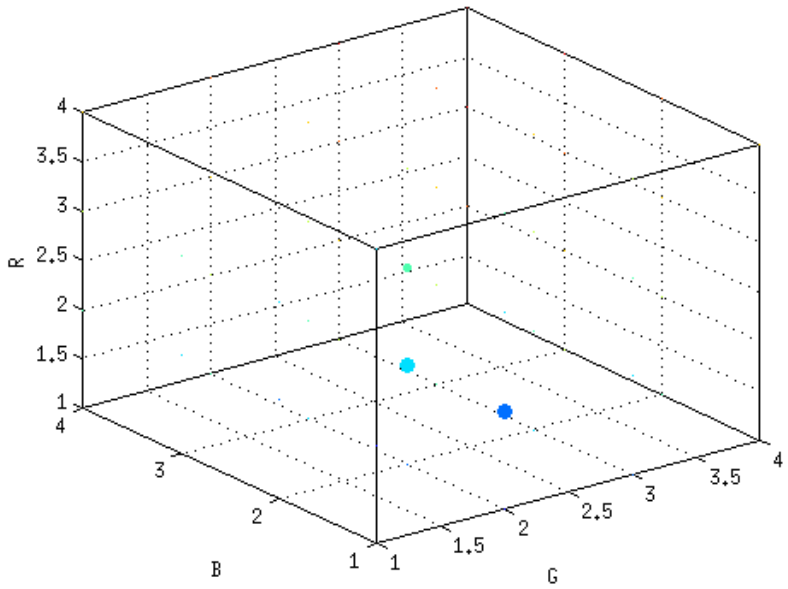

Figura 4.22: Histograma 3D das bordas do bloco de 32 × 32 *pixels*, externo a face do quadro Claire.

Tabela 4.2: Algoritmo Blocos Particionados.

$k = 0$ (Inicialização)

$n \leftarrow 32$: Dimensão do bloco em *pixels*

$bins_R = bins_G = bins_B \leftarrow 4$: *Bins* para cada canal do quadro RGB

d_T : Distância de *threshold* utilizada para calcular os blocos válidos

No quadro z_0 define a região da face com z_{model}

$z_{model} \leftarrow z_0$

z_{masc} : máscara binária associada à região da face no quadro z_{model}

Calcula h_{model} utilizando a Equação (4.10)

Normaliza h_{model}: $h_{model;normalizado} \leftarrow \frac{h_{model}}{\sum_i h_{model}}$

Para cada novo quadro z_k:

$b_k^{i,j}$: Bloco $n \times n$ interno ao quadro z_k, Figura 4.14

$bb_k^{i,j}$: *pixels* da borda de $b_k^{i,j}$

$hbb_k^{i,j}$: Histograma 3D de $bb_k^{i,j}$ calulado a partir da Equação (4.10)

Compara histogramas da bordas de cada bloco com o modelo através da distância de Bhattacharyya $d_k^{i,j}$ definida na Equação (4.13):

FOR $i = 1 : max_v$

FOR $j = 1 : max_h$

$$fb_k^{i,j} = \begin{cases} 1, & d_k^{i,j} \leq d_T, \text{ é bloco válido} \\ 0, & \text{outros valores} \end{cases}$$

END-FOR

END-FOR

Retorna fb_k, uma imagem binária contendo as regiões no quadro z_k que pertencem ao objeto (face) a ser acompanhado.

Quanto a complexidade computacional envolvida em função da resolução do vídeo de entrar. Dado uma resolução para cada quadro do vídeo definido por $\dim(\mathbf{z}_k) = [n_x\ n_y]$, n_x e n_y múltiplos de 32, com uma complexidade computacional capaz de tratar o total de blocos 32×32 na ordem de $N(O) = \frac{n_x}{32} \times \frac{n_y}{32}$. Um aumento na resolução do quadro para $\dim(\mathbf{z}'_k) = [n_x + A\ n_y + B]$, sendo A e B múltiplos de 32, impõe um aumento da complexidade para $N(O)' = N(O) + \frac{B\ n_x}{32} + \frac{A\ n_y}{32} + \frac{A\ B}{32}$.

4.4.4 *Particle likelihood*

A definição do *importance function* baseado na estimativa de deslocamento da região de acompanhamento sugere a localização mais provável da face no instante corrente k. Esta localização será utilizada pelo algoritmo de processamento do filtro para gerar as partículas $\mathbf{X}_k^i$. Cada amostra $\mathbf{X}_k^i$ precisa ser validada com o quadro $\mathbf{z}_k$, de modo que seja possível calcular o peso ω_k^i, que significa a importância que cada partícula possui na definição da estimativa do vetor de estado $\hat{\mathbf{x}}_k$.

A definição de ω_k^i para cada amostra $\mathbf{X}_k^i$ é obtida a partir da sua informação de distribuição marginal de verossimilhança $p(\mathbf{z}_k|\mathbf{x}_k)$. Duas importantes hipóteses podem ser calculadas a partir da elipse que é representada parametricamente por cada amostra $\mathbf{X}_k^i$, como desenvolvido por [BS04]:

- a informação de gradiente da borda da elipse, ν_{borda}^i;
- a informação de histograma de cor do interior da elipse, ν_{cor}^i.

O gradiente da borda de $\mathbf{X}_k^i$, ν_{borda}^i, informa qual percentual do contorno da elipse foi detectado sob a região de transição da face do interlocutor com o plano de fundo da cena. Uma elipse que representa melhor o contorno da face terá maior informação de gradiente detectado.

Para o histograma de cor do interior da elipse, ν_{cor}^i, quanto menor a diferença entre a distribuição dos *bins* que representam a informação de cor da região no quadro $\mathbf{z}_k$ do interior da elipse em relação ao conjunto de *bins* do histograma do modelo da face $H_M(x, y, z)$, como definido na Equação (4.10), maior a probabilidade deste vetor de estado representar a localização da face no quadro $\mathbf{z}_k$.

Adicionalmente, baseado na informação de módulo de cor e de gradiente do vetor $\mathbf{X}_k^i$ (gradiente e histograma), ambas as informações utilizadas para o cálculo da verossimi-

lhança são consideradas condicionalmente independentes entre si. Dessa forma, é possível representar, para cada amostra aleatória $\mathbf{X}_k^i$, um índice ν_k^i que representa a distribuição da verossimilhança no tempo k, definido como

$$p(\mathbf{z}_k|\mathbf{X}_k^i) = \nu_k^i = \nu_{borda}^i \, \nu_{cor}^i \quad . \tag{4.17}$$

A descrição do algoritmo proposto para o cálculo de ν_{borda}^i e ν_{cor}^i é apresentada a seguir.

Módulo de cor

O módulo de cor ν_{cor}^i segue a mesma metodologia aplicada aos blocos particionados, apresentada na Seção 4.4.3. Entretanto, todos os *pixels* do interior da elipse definida pela amostra $\mathbf{X}_k^i$ são considerados no cálculo do histograma.

Definindo $H(\mathbf{X}_k^i)$ o histograma de três dimensões da partícula $\mathbf{X}_k^i$ e já contando com o histograma modelo dado pela Equação (4.10), $H_M(\mathbf{x})$, a distância $d_B(H_M(\mathbf{x}), H(\mathbf{X}_k^i))$ é calculada por

$$d_B(H_M(\mathbf{x}), H(\mathbf{X}_k^i)) = \sqrt{1 - \sum_i \frac{\sqrt{H_M^i(\mathbf{x}) H(\mathbf{X}_k^i)}}{\sqrt{\sum_i H_M^i(\mathbf{x}) \sum_i H(\mathbf{X}_k^i)}}} \, , \tag{4.18}$$

que é idêntica à Equação (4.12), apenas substituindo o parâmetro $H_{borda}(b_{i,j}^k)$ por , $H(\mathbf{X}_k^i)$.

O valor associado ao módulo de cor para a partícula $\mathbf{X}_k^i$, no instante de tempo k, é definido por

$$\nu_{cor}^i = \sqrt{\frac{1}{2 \, \pi \, \sigma_{cor}^2}} \exp\left(\frac{-d_B^2}{2\sigma_{cor}^2}\right) \quad , \tag{4.19}$$

onde σ_{cor}^2 é a variância definida para esta distribuição Gaussiana.

Módulo gradiente

O módulo gradiente é calculado a partir de um conjunto de N_{ell} *pixels* $\{\mathbf{p}_j^i, \; j = 1, 2, ..., N_{ell}\}$ pertencentes ao contorno da elipse parametrizada $\mathbf{X}_k^i$.

Para cada *pixel* $\mathbf{p}_j^i$ é calculado o gradiente do vetor normal ao contorno da elipse

$$\nabla \mathbf{p} \equiv grad(\mathbf{p}) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix} \quad , \tag{4.20}$$

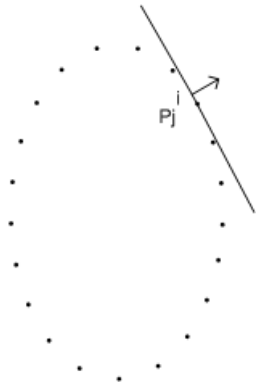

Figura 4.23: O módulo gradiente é obtido a partir do gradiente do vetor normal de conjunto de pontos localizados no contorno da elipse.

Figura 4.23, localizado nas coordenadas (x, y) do plano $\mathbf{S}$ do quadro $\mathbf{z}_k$ [GW08].

Por sua vez, a magnitude $M(x, y)$ do vetor $\nabla\mathbf{p}$ é definida como

$$M(x, y) = mag(\nabla\mathbf{p}) = \sqrt{g_x^2 + g_y^2} \quad . \tag{4.21}$$

Considerando cada *pixel* $\mathbf{p}_j^i$ pertencente à fronteira da elipse $\mathbf{X}_k^i$, o módulo gradiente $\phi_g(\mathbf{X}_k^i)$ é calculado a partir do somatório de cada magnitude $M(x_j, y_j)$ dos $N_{\mathbf{X}_k^i}$ *pixels* que compõem a fronteira de $\mathbf{X}_k^i$, conforme

$$\phi_g(\mathbf{X}_k^i) = \frac{1}{N_{\mathbf{X}_k^i}} \sum_{j=1}^{N_{\mathbf{X}_k^i}} M(x_j, y_j) \quad . \tag{4.22}$$

Normalizando o valor apresentado na Equação (4.22), para que seja possível calcular o índice ν_k^i, definido na Equação (4.17), deve-se converter o valor integral $\phi_g(\mathbf{X}_k^i)$ em um valor percentual entre 0 e 1. Esse índice é definido por

$$\nu_{borda}^i = \frac{\phi_g(\mathbf{X}_k^i) - \min_{\mathbf{X}_k^i}\{\phi_g(\mathbf{X}_k^i)\}}{\max_{\mathbf{X}_k^i}\{\phi_g(\mathbf{X}_k^i)\} - \min_{\mathbf{X}_k^i}\{\phi_g(\mathbf{X}_k^i)\}} \quad , \tag{4.23}$$

onde os operadores $min\{.\}$ e $max\{.\}$ representam as componentes de gradiente de menor e maior valor, respectivamente, calculados para o vetor de estado $\mathbf{X}_k^i$.

4.4.5 Modelo de transição do estado da face

Como apresentado por [BS04], a transição da face entre quadros na sequência de vídeo pode ser modelada como um processo auto-regressivo de primeira ordem, dado por

$$\mathbf{x}_k = \mathbf{A}\ \mathbf{x}_{k-1} + \mathbf{B}\ \omega_k \tag{4.24}$$

e

$$\omega_k = \mathcal{N}(0, \Sigma_p) \quad , \tag{4.25}$$

onde ω_k é modelado como um ruído gaussiano de média zero e variância Σ_p e as matrizes $\mathbf{A}$ e $\mathbf{B}$ representam, respectivamente, as componentes determinística e estocástica do modelo de predição apresentado, podendo ser consideradas como matrizes identidades, $\mathbf{A} = \mathbf{I}$ e $\mathbf{B} = \mathbf{I}$.

A partir da Equação (4.24), a distribuição de probabilidade da transição de estados da face numa sequência de vídeo, $p(\mathbf{x}_k = \mathbf{X}_k^i | \mathbf{x}_{k-1} = \mathbf{X}_{k-1}^i)$, pode ser calculada por

$$p(\mathbf{x}_k = \mathbf{X}_k^i | \mathbf{x}_{k-1} = \mathbf{X}_{k-1}^i) \equiv \mathcal{N}(\mathbf{X}_{k-1}^i, \Sigma_p) \quad , \tag{4.26}$$

onde o índice $i = 1, \cdots, N_p$ representa cada partícula definida no filtro.

4.4.6 Descrição do algoritmo de acompanhamento da face

Com todas as distribuições de probabilidades modeladas, é necessário, agora, construir uma descrição consolidada do algoritmo de acompanhamento da face, utilizando a metodologia *Particle Filter* empregada neste trabalho. Na Tabela 4.3, é apresentado o algoritmo utilizado para implementar o acompanhamento da face do interlocutor.

Partindo da demonstração, definida pela Equação (2.66), de como é possível calcular numericamente o peso ω_k^i de cada partícula definida pelo *Particle Filter*, adicionado ao conhecimento sobre a transição entre os estados $\mathbf{x}_k | \mathbf{x}_{k-1}$ de cada partícula, visto que uma nova observação $\mathbf{z}_k$ foi realizada no tempo k, é possível definir uma solução, a partir da Equação (4.27), para o cálculo do peso ω_k^i, indicando qual a importância da amostra do par $\{\mathbf{X}_k^i | \omega_k^i\}$ na definição da estimação da densidade de probabilidade *a posteriori* marginal $\hat{p}(\mathbf{x}_k | \mathbf{z}_k)$. Tal peso é calculado por

$$\omega_k^i = \frac{\sum_{j=1}^{N_p} w_{k-1}^j\ p(\mathbf{X}_k^i | \mathbf{X}_{k-1}^j)}{q(\mathbf{X}_k^i)}\ p(\mathbf{z}_k | \mathbf{X}_k^i) \quad . \tag{4.27}$$

Tal como foi definida na Tabela 4.3, a estimação do vetor de estados, no tempo k, pode ser calculada como o valor médio da densidade de probabilidade *a posteriori* marginal estimada, dado por

$$\hat{\mathbf{x}} = E\{\hat{p}(\mathbf{x}_k | \mathbf{z}_k)\} \quad . \tag{4.28}$$

Tabela 4.3: Algoritmo de acompanhamento da face.

$k = 0$ (Inicialização)

$\mathbf{z}_0$ foi armazenado

$N_p \leftarrow 100$

$N_r \leftarrow 1$

$x_0 \leftarrow$ {seleção da face do interlocutor}

FOR $i \leftarrow 1 : N_p$

$\mathbf{X}^i_{k=0} \leftarrow \mathcal{N}(x_0, \Sigma_G)$

$\mathbf{W}^i_{k=0} \leftarrow \frac{1}{N_p}$

END-FOR

LOOP

$\{\mathbf{X}^i_{k-1}, \mathbf{W}^i_{k-1}\} \leftarrow \{\mathbf{X}^i_k, \mathbf{W}^i_k\}$

$\mathbf{z}_k$ foi armazenado

$\Delta\mathbf{X}_k \leftarrow$ Blocos Particionados, Tabela 4.2

$\mathbf{X}^i_k \leftarrow$ Equação (4.4)

$q(\mathbf{X}_k|\mathbf{z}_k) \leftarrow$ Equação (4.5)

$p(\mathbf{X}^i_k|\mathbf{X}^i_{k-1}) \leftarrow$ Equação (4.26)

$p(\mathbf{z}_k|\mathbf{X}^i_k) \leftarrow$ Equação (4.17)

$\omega^i_k \leftarrow$ Equação (4.27)

$\mathbf{W}^i_k \leftarrow$ Equação (2.67)

$N_{eff} \leftarrow$ Equação (2.70)

$\hat{\mathbf{X}}_k \leftarrow$ Equação (4.28)

IF $N_{eff} \ll N_T$ **THEN** Aplica *resampling*

$k \leftarrow k + 1$

END-LOOP

Capítulo 5

Estabilização do vídeo digital

5.1 Introdução

A metodologia adotada utilizando os fundamentos do *Particle Filter*, como demonstrado no algoritmo apresentado na Seção 4.4.6, identifica a estimativa de movimento inter-quadros da face do interlocutor, definindo um novo estado para o vetor $\mathbf{x}_k$. Para obter a estabilização do vídeo, com o resultado esperado, deve-se aplicar uma transformação $\mathbf{T}\{.\}$ ao quadro corrente $\mathbf{z}_k$, de modo a minimizar a vibração descrita no Capítulo 1.

Nesse capítulo, será descrito como resolver o problema da minimização da vibração inter-quadros, possibilitando uma transição suave na sequência de vídeo digital.

5.2 Estabilização do vídeo a partir do vetor de estados

A estabilização do vídeo deve compensar o movimento indesejado, ocorrido na câmera quadro a quadro. Assumindo que o deslocamento $\Delta\mathbf{x}_k$ do vetor de estados $\mathbf{x}_k$ foi puramente intencional, a compensação de movimento da face inter-quadros deve seguir o modelo

$$\begin{bmatrix} z_{x,k} \\ z_{y,k} \end{bmatrix} = \begin{bmatrix} \cos(\Delta\mathbf{x}_k(\phi)) & -\sin(\Delta\mathbf{x}_k(\phi)) \\ \sin(\Delta\mathbf{x}_k(\phi)) & \cos(\Delta\mathbf{x}_k(\phi)) \end{bmatrix} \begin{bmatrix} z_{x,k-1} \\ z_{y,k-1} \end{bmatrix} + \begin{bmatrix} T_k(\Delta\mathbf{x}_k(x_c)) \\ T_k(\Delta\mathbf{x}_k(y_c)) \end{bmatrix} \quad , \tag{5.1}$$

onde o par $(z_{x,k}, z_{y,k}) \in \mathcal{S}$ são as coordenadas de cada *pixel* do quadro $\mathbf{z}_k$, no tempo k, pertencentes ao plano $\mathcal{S}$, $\Delta\mathbf{x}_k(\phi)$ é o parâmetro de deslocamento angular da face

previsto na Equação (3.6) e o par $(T_k(\Delta \mathbf{x}_k(x_c)), T_k(\Delta \mathbf{x}_k(y_c)))$ são as translações no eixo horizontal e vertical, respectivamente, que compensam o deslocamento estimado da face do interlocutor.

Para resolver o problema da estabilização do vídeo, como proposto neste trabalho, é necessário acompanhar somente o movimento de translação da face, ou seja, o movimento de deslocamento da face projetada no plano $\mathcal{S}$, gerado a cada intervalo de tempo discreto. O plano $\mathcal{S}$ representa o plano de duas dimensões da câmera onde a cena de três dimensões está projetada [Tek95], assim como ilustrado na Figura 5.1. A Figura 5.2 ilustra o deslocamento da face representando a sua translação e rotação no plano $\mathcal{S}$ entre os quadros $\mathbf{z}_{k-1}$ e $\mathbf{z}_k$, respectivamente.

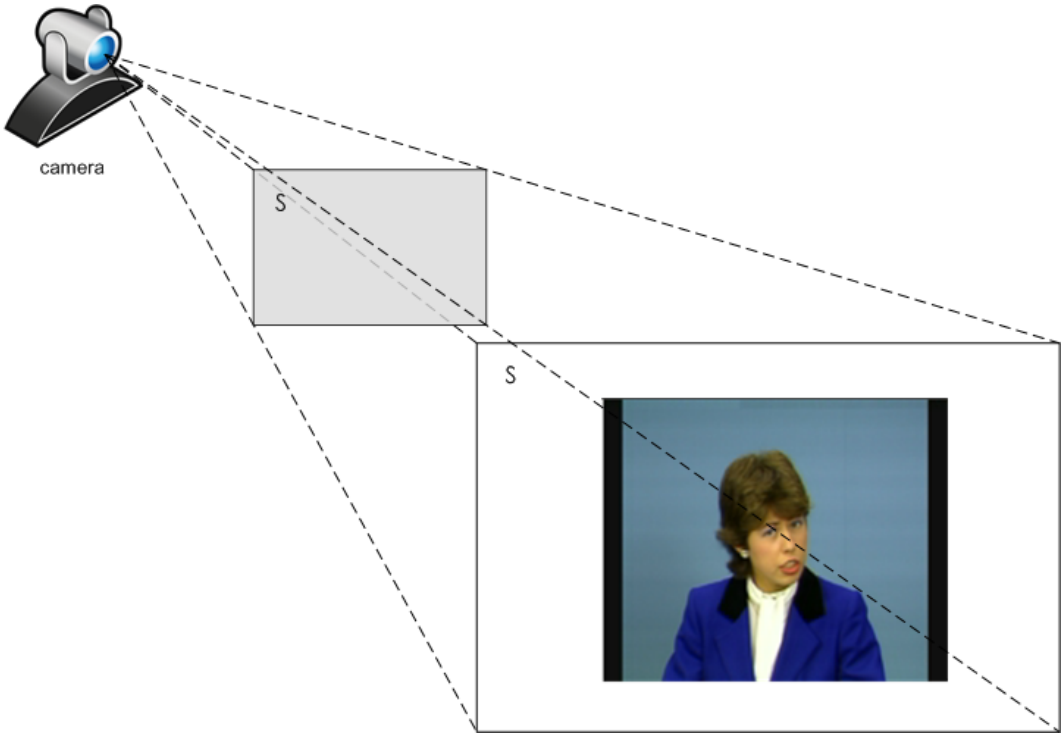

Figura 5.1: Projeção da image 3D no plano da câmera 2D. Em todas as transformações aplicadas aos quadros de vídeo serão utilizados a projeção 2D, ou seja, o plano S.

O algoritmo de estabilização fará uso de uma janela $\mathbf{P}$ que desliza pelo plano do quadro $\mathcal{S}$, $\mathbf{P} \subset \mathcal{S}$, de modo a minimizar o erro entre o centro da janela $\mathbf{P}$ e o centro da elipse $(x_c, y_c)_k$ pertencente ao vetor de estados $\mathbf{x}_k$. Assim, podemos definir um vetor $\mathbf{p}_k$ que representa a posição da janela $\mathbf{P}$ sobre o plano $\mathcal{S}$, no instante de tempo k, conforme

$$\mathbf{p}_k = [\ x_o \quad y_o \quad w \quad h\ ]^T \quad , \tag{5.2}$$

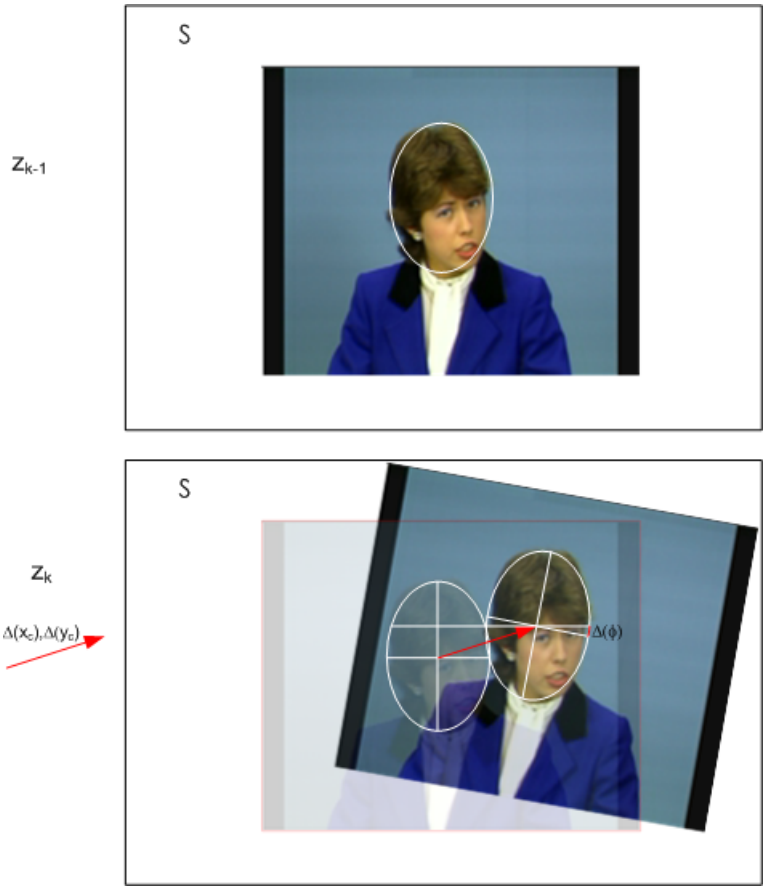

Figura 5.2: Movimento de translação e rotação da face no plano S, ocorrido entre dois quadros consecutivos.

onde (x_o, y_o) é a coordenada de origem do plano $\mathbf{P}$ no plano $\mathcal{S}$ que localiza a posição inicial desta janela; (w, h) são as dimensões horizontal e vertical da janela $\mathbf{P}$ respectivamente. A Figura 5.3 ilustra os elementos do vetor $\mathbf{p}_k$ na definição da janela $\mathbf{P}$ dentro do plano $\mathbf{S}$.

Nesse esquema, é possível identificar o centro da janela $\mathbf{P}$ como

$$x_c = x_o + \frac{w}{2} \tag{5.3}$$

$$y_c = y_o + \frac{h}{2} \ . \tag{5.4}$$

De (5.3) e (5.4), podemos definir um novo vetor indicando a localização do centro do plano $\mathbf{P}$ no eixo de coordenadas da plano $\mathbf{S}$, dado por

$$\mathbf{P}_{centro,k} = [\ x_c \quad y_c\ ]^T \quad . \tag{5.5}$$

O esquema adotado de translação da janela $\mathbf{P}$, no tempo k, pode ser tratado como um processo auto-regressivo de primeira ordem e definido por

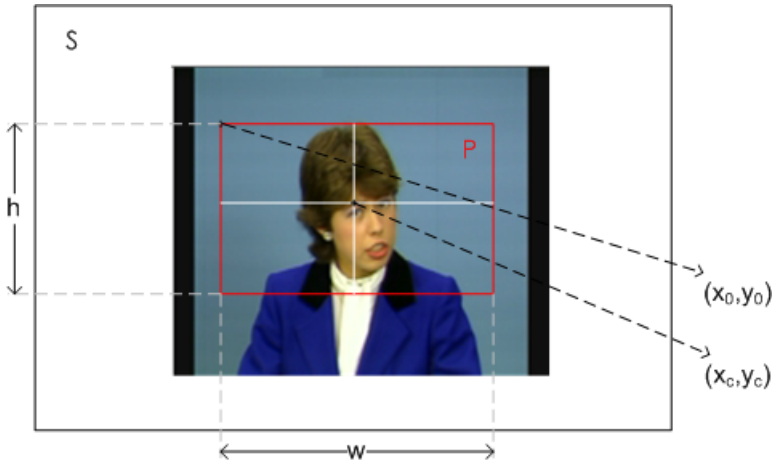

Figura 5.3: Plano P representando a janela deslizante sobre o plano da imagem S.

$$\mathbf{p}_k + \alpha\ \mathbf{p}_{k-1} = \beta\ \mathbf{x}_k \quad . \tag{5.6}$$

A distância $\mathbf{d}_k$ entre as coordenadas da elipse, no tempo k, $\mathbf{xc}_k = (x_c, y_c)_k$, e a coordenada do centro da janela $\mathbf{P}$ em $k-1$, $\mathbf{p}_{k-1}$ é dada por

$$\mathbf{d}_k = \mathbf{xc}_k - \mathbf{p}_{k-1} \quad . \tag{5.7}$$

A nova posição da janela $\mathbf{P}$ em k, $\mathbf{p}_k$, pode ser definida também em função de $\mathbf{d}_k$, como

$$\mathbf{p}_k = \mathbf{p}_{k-1} - \mathbf{A}\ \mathbf{d}_k \quad . \tag{5.8}$$

Substituindo (5.7) em (5.8), temos

$$\begin{aligned} \mathbf{p}_k &= \mathbf{p}_{k-1} + \mathbf{A}\ (\mathbf{x}_k - \mathbf{p}_{k-1}) \\ \mathbf{p}_k &= (1 - \mathbf{A})\ \mathbf{p}_{k-1} + \mathbf{A}\ \mathbf{x}_k \\ \mathbf{p}_k &+ (\mathbf{A} - 1)\ \mathbf{p}_{k-1} = \mathbf{A}\ \mathbf{x}_k \quad . \end{aligned} \tag{5.9}$$

Comparando as Equações (5.6) com (5.9), temos os valores dos coeficientes α e β, em função da matriz 2×2 de ajuste $\mathbf{A}$, dados por

$$\begin{aligned} \alpha &= \mathbf{A} - 1 \\ \beta &= \mathbf{A} \quad , \end{aligned} \tag{5.10}$$

onde **A** é definida como

$$\mathbf{A} = \begin{bmatrix} a_1 & 0 \\ 0 & a2 \end{bmatrix} \quad . \tag{5.11}$$

Tratando o deslocamento da janela **P** no plano do quadro $\mathcal{S}$ tal como definido pela Equação 5.9), é possível implementar um algoritmo com uma resposta suave, que irá convergir o centro da janela **P** para a posição estimada do centro da face, evitando uma movimentação brusca da janela **P** e, assim, promovendo a qualidade final do acompanhamento em tempo-real da face do interlocutor. Uma vantagem desta proposta adotada é permitir um processamento simples e rápido da posição da janela **P**, a cada intervalo de tempo, evitando utilizar uma transformação *affine pixel* a *pixel* de cada quadro como proposto por [Yan09].

Capítulo 6

Resultados experimentais

6.1 Introdução

Este capítulo é dedicado à comprovação das técnicas apresentadas nos Capítulos 3, 4 e 5. Serão apresentados os cenários utilizados como teste e os resultados obtidos para cada uma das técnicas expostas. Os algoritmos Módulo de Cor e Módulo Gradiente são apresentados inicialmente, por serem técnicas selecionadas de artigos similares. Entretanto, eles receberam uma implementação adequada à sua utilização nesse trabalho. Em seguida, a técnica Blocos Particionados (BP) recebe uma atenção mais detalhada, devido à sua abordagem singular e à sua proposta inovadora. As três técnicas combinadas neste trabalho tentam oferecer uma abordagem à metodologia PF, tendo em mente aplicações em tempo real. Por fim, são apresentados os resultados obtidos pelo algoritmo de estabilização do vídeo, por ser tratar do objetivo a ser alcançado com as técnicas propostas.

6.2 Resultados para o algoritmo Módulo de Cor

Como apresentado na Seção 4.4.4, o algoritmo de Módulo de Cor constrói um histograma 3D da distribuição de intensidade dos *pixels* dos três canais de cor (RGB) da região interna à elipse que identifica cada uma das partículas pertencentes ao *Particle Filter*. Aqui são apresentados alguns resultados obtidos na utilização desse algoritmo.

Para demonstrar o funcionamento do algoritmo Módulo de Cor, foi utilizado o conjunto de imagens estáticas, apresentado na Tabela 6.1, que comprovam seu correto funcionamento. Todas essas imagens possuem três canais de cor do tipo RGB.

Tabela 6.1: Relação das imagens estáticas utilizadas para apresentação dos resultados esperados do algoritmo Módulo de Cor.

Nome do Quadro	Nome do Arquivo	Comentário
NEGRO	BLACK.png	Imagem de dimensão 640 × 480 *pixels*, onde todos os *pixels* possuem valor zero.
BRANCO	WHITE.png	Imagem de dimensão 640 × 480 *pixels*, onde todos os *pixels* possuem valor 255.
AZUL	BLUE.png	Imagem de dimensão 480 × 480 *pixels*, onde somente a componente de cor azul da imagem possui valor 255, enquanto as outras duas componentes possuem valor zero.
VERMELHO	RED.png	Imagem de dimensão 480 × 480 *pixels*, onde somente a componente de cor vermelha da imagem possui valor 255, enquanto as outras duas componentes possuem valor zero.
VERDE	GREEN.png	Imagem de dimensão 480 × 480 *pixels*, onde somente a componente verde da imagem possui valor 255, enquanto as outras duas componentes possuem valor zero.
Lena	Lena.jpg	Imagem Lena, ilustrada na Figura D.1.
Alcatraz	Alcatraz.jpg	Imagem Alcatraz, ilustrada na Figura D.2.

A primeira análise utiliza, em conjunto, os quadros BRANCO e NEGRO para comprovar o dois casos extremos do histograma gerado pelo algoritmo Módulo de Cor, como pode ser visto na Figura 6.1. Observando-se, no plano tridimensional, a distribuição do histograma do quadro BRANCO, pode-se confirmar que todos os *pixels* encontram-se no *bin* representado pela posição $[5, 5, 4]$, ilustrado pela região em vermelho. Por outro lado, o histograma gerado a partir do quadro NEGRO possui todos os seus *pixels* localizado no *bin* $[5, 5, 1]$. Ambos os planos apresentados representam os limites superior e inferior, válidos dentro do espaço tridimensional, para qualquer imagem utilizada nesse algoritmo.

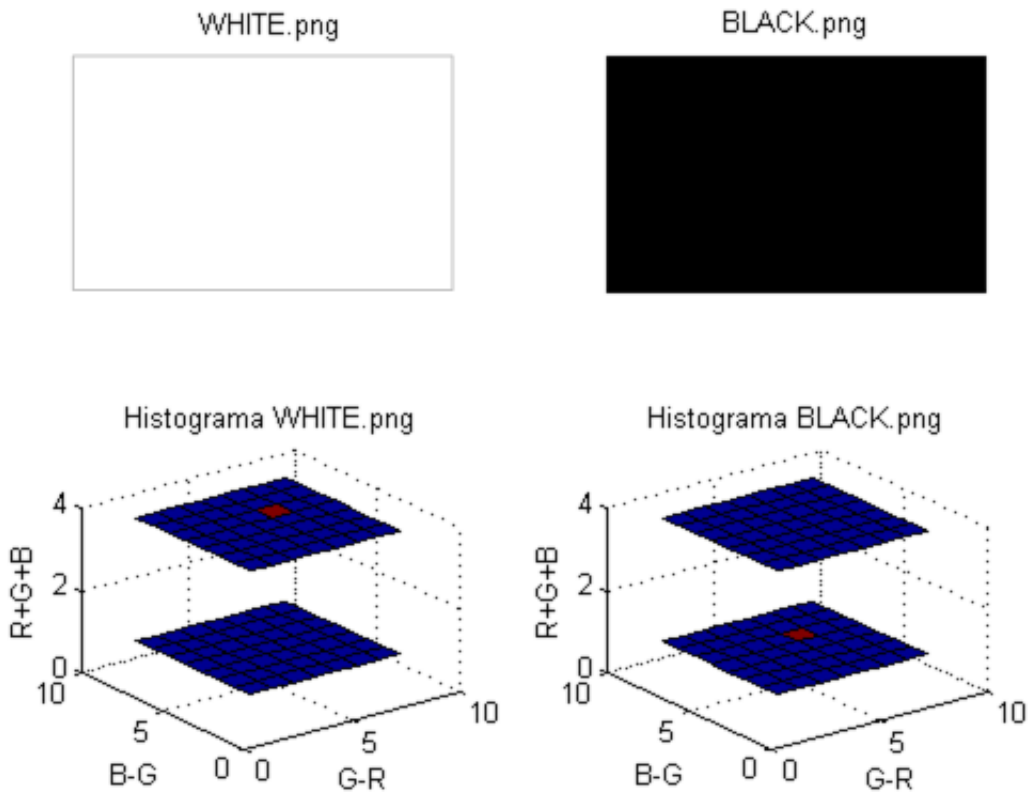

Figura 6.1: Comparativo entre histogramas gerados pelo algoritmo Módulo de Cor quando a imagem é composta por *pixels* de valor zero (quadro NEGRO) e quando a imagem é composta por *pixels* de valor 255 (quadro BRANCO): (a) Quadro branco (WHITE.png). (b) Quadro negro (BLACK.png). (c) Plano superior localiza o *bin* do histograma de cor onde os *pixels* do quadro BRANCO estão localizados. (d) Plano superior localiza o *bin* do histograma de cor onde os *pixels* do quadro NEGRO estão localizados.

Tomando-se os quadros VERMELHO, VERDE e AZUL, em conjunto, e utilizando-se a mesma análise realizada com os quadros BRANCO e NEGRO, podem-se visualizar as regiões representativas das cores fundamentais utilizadas para cada canal RGB dentro do espaço tridimensional definido para o algoritmo. Analisando a Figura 6.2 é possível verificar o comportamento de cada componente de cor dentro do espaço tridimensional.

A análise conjunta dos resultados obtidos, apresentados nas Figuras 6.1 e 6.2 permite definir o espaço de cor resultante da transformação linear do modelo de cor RGB [GW08] através das transformações lineares definida pela Equação (4.7). Na Tabela 6.2, é apresentada a relação entre os valores de *pixels* dos quadros NEGRO, BRANCO, VERMELHO, VERDE e AZUL, e suas respectivas transformações.

Tabela 6.2: Aplicando a transformação linear da Equação (4.7) aos quadros usados no exemplo.

Nome do Quadro	{R,G,B}	{B-G,G-R,R+G+B}	{bins}
NEGRO	{ 0, 0, 0}	{ 0, 0, 0}	{4,4,1}
BRANCO	{255,255,255}	{ 0, 0,765}	{4,4,4}
VERMELHO	{255, 0, 0}	{ 0,-255,255}	{4,1,2}
VERDE	{ 0,255, 0}	{-255, 255,255}	{1,8,2}
AZUL	{ 0, 0,255}	{ 255, 0,255}	{8,4,2}

Partindo para a análise do algoritmo implementando a técnica Módulo de Cor, serão utilizada as imagens “Lena”, “Claire” e “Alcatraz”. Para cada uma das imagens, serão apresentados a região utilizada como modelo, selecionada por um retângulo em cor azul, enquanto que a região utilizada como referência para comparar com o modelo é selecionada por um retângulo branco. Não necessariamente os retângulos possuem tamanhos idênticos, visto que as partículas criadas aleatoriamente podem possuir diferentes tamanhos.

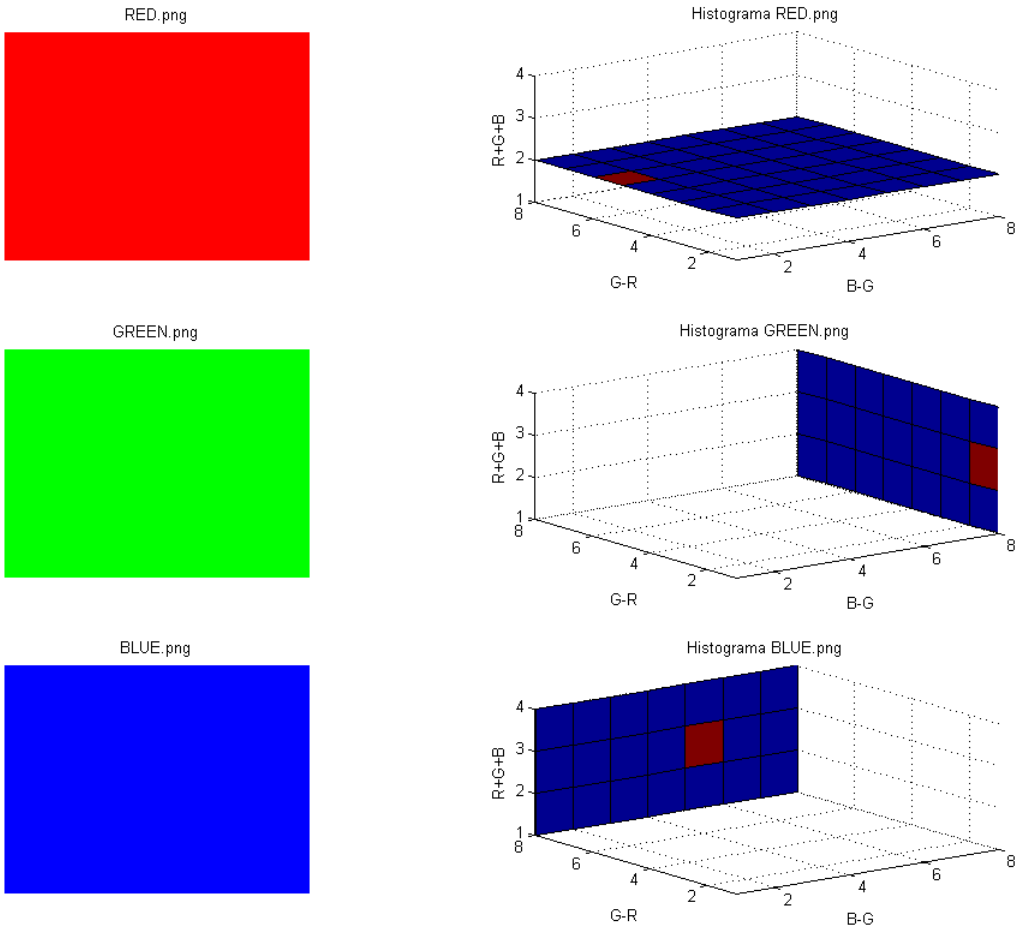

Figura 6.2: Comparativo entre histogramas gerados pelo algoritmo Módulo de Cor quando a imagem é composta por *pixels* com os valores fundamentais de cor (vermelho, verde e azul): (a) Quadro VERMELHO (red.png). (b) Plano no histograma de cor onde estão localizadas todas as ocorrências do *pixel* de cor vermelho $[R, G, B] = [255, 0, 0]$. (c) Quadro VERDE (green.png). (d) Plano no histograma de cor onde estão localizadas todas as ocorrências do *pixel* de cor verde $[R, G, B] = [0, 255, 0]$. (e) Quadro AZUL (blue.png). (f) Plano no histograma de cor onde estão localizadas todas as ocorrências do *pixel* de cor azul $[R, G, B] = [0, 0, 255]$.

Nas Figuras 6.3, 6.4 e 6.5, são mostrados três casos de uso do cálculo do Módulo de Cor utilizando a mesma região selecionada como modelo mas diferentes regiões de referência. Como esperado, quando a região modelo é idêntica à região de referência, a distância Bhattacharyya entre os histogramas tem distância igual ao valor unitário, significando que ambos os histogramas possuem a mesma distribuição entre os *bins*, $d_B = 1,0$. Esse caso é ilustrado na Figura 6.3.

Figura 6.3: Distância Bhattacharrya aplicada à imagem Lena, onde a região modelo é idêntica à região de referência. Distância = 1,0.

Quando uma parte da região selecionada definida como modelo sobrepõe a região selecionada como referência, como no caso da Figura 6.4, a distância calculada entre os histogramas torna-se menor do que um, $d_B < 1,0$.

Os casos em que se obtêm os menores valores para a distância Bhattacharyya ocorrem quando a região do modelo difere completamente da região de referência, não havendo nenhuma sobreposição, tal como apresentado na Figura 6.5. Neste caso, os valores da distância são ainda maiores que no caso anterior, $d_B \ll 1,0$.

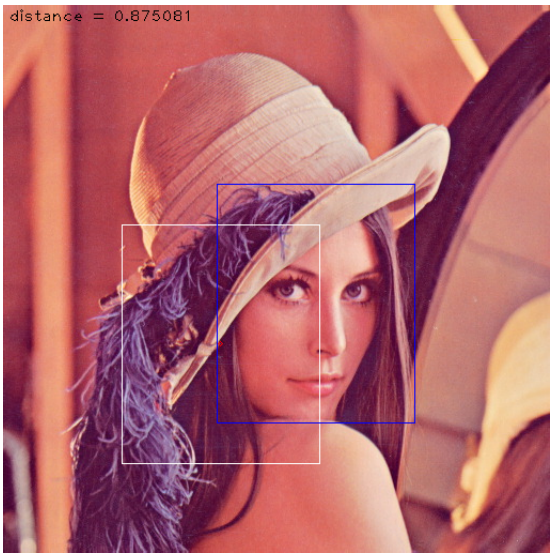

Figura 6.4: Distância Bhattacharrya aplicada à imagem Lena, onde a região modelo sobrepõe parte da região de referência. Distância = 0,875081.

Figura 6.5: Distância Bhattacharrya aplicada à imagem Lena, onde a região modelo e a região de referência são distintas. Distância = 0,803027.

A imagem Alcatraz, Figura D.2, possui um cenário bastante complexo, com o fundo contendo diversos detalhes, o que é propício para a análise do algoritmo Módulo de Cor. As Figuras 6.6 a 6.10 apresentam diferentes regiões de referência para a mesma região modelo, a qual compreende a face selecionada.

Figura 6.6: Distância Bhattacharyya aplicada à imagem Alcatraz, onde a região modelo é idêntica à região de referência. Distância = 1,0.

Figura 6.7: Distância Bhattacharyya aplicada à imagem Alcatraz, onde a região modelo sobrepõe parte da região de referência. Distância = 0,741985.

Figura 6.8: Distância Bhattacharyya aplicada à imagem Alcatraz, onde a região modelo e a região de referência são distintas. Distância = 0,392631.

Figura 6.9: Distância Bhattacharyya aplicada à imagem Alcatraz, onde a região modelo sobrepõe parte da região de referência. Distância = 0,609899.

Figura 6.10: Distância Bhattacharyya aplicada à imagem Alcatraz, onde a região modelo e a região de referência são próximas, porém distintas. Distância = 0,412853.

6.3 Resultados para o algoritmo Módulo Gradiente

A Seção 4.4.4 apresentou como o algoritmo *Particle Filter* implementado neste trabalho identifica a região de borda da elipse de cada uma das $n = 1, \ldots, N_p$ amostras aleatórias $\mathbf{X}_k^n$, no tempo k. Os experimentos a seguir mostram os resultados obtidos para a imagem Alcatraz. As imagens coloridas, contendo os três canais de cores R, G e B são convertidas para o canal de cor monocromático de intensidade. Esta conversão é necessária pois somente o gradiente de intensidade é considerado neste algoritmo.

Na Figura 6.11, é indicada a região selecionada (retângulo em branco) na imagem Alcatraz utilizada para analizar os resultados do algoritmo. A Figura 6.12 apresenta o detalhe da região selecionada. A imagem da esquerda apresenta os locais onde o gradiente é utilizado para detectar o contorno da face, enquanto que a imagem da direita apresenta uma elipse e os gradiente utilizados para a detecção do contorno da face.

Figura 6.11: Módulo Gradiente calculado para a área selecionada na imagem Alcatraz que compreende a face. Gradiente = 0,0392616.

A Figura 6.13 seleciona uma região da imagem onde a elipse de busca está parcialmente deslocada em relação à face. A Figura 6.14 apresenta, em detalhe, a região selecionada e os gradientes utilizados para o cálculo do contorno da elipse.

Figura 6.12: Detalhe da elipse selecionada na Figura 6.11: (a) Gradiente da região selecionada. (b) Gradientes utilizados para o cálculo da região de contorno definida pela elipse. Quanto mais próximos os gradientes utilizados neste cálculo indica que a elipse possivelmente define a região de contorno da face do interlocutor.

Figura 6.13: Módulo Gradiente calculado para a área selecionada na imagem Alcatraz que parcialmente seleciona a face. Gradiente = 0,0390619.

Figura 6.14: Detalhe da elipse selecionada na Figura 6.13: (a) Gradiente da região selecionada. (b) Gradientes utilizados para o cálculo da região de contorno definida pela elipse.

6.4 Resultados do algoritmo Blocos Particionados

Como apresentado na Seção 4.4.3, a técnica Blocos Particionados (BP), que implementa a *importance function*, utilizada pelo *particle function*, divide cada quadro do vídeo em blocos de 32×32 *pixels* e localiza, a partir do modelo previamente definido, quais blocos pertencem à região da face do interlocutor.

Utilizando o quadro Alcatraz como referência para esta experimentação, a Figura 6.15 apresenta a imagem segmentada em diversos blocos de 32×32 *pixels*, que serão utilizados para a definição da região na imagem onde se encontra a face do interlocutor, que é o objeto de acompanhamento selecionado.

Figura 6.15: Imagem Alcatraz dividida em blocos de dimensão 32×32 *pixels* para iniciar o processamento do algoritmo Blocos Particionados. A região utilizada como modelo pelo algoritmo está indicada pelo retângulo em azul na região da face.

O histograma de cor de cada bloco identificado na Figura 6.15 será comparado ao histograma utilizado como modelo, que, no caso deste experimento, é a região retangular selecionada em azul. As Figuras 6.16 e 6.17 apresentam, respectivamente, os resultados da comparação entre cada bloco e o modelo utilizando um *threshold* inferior igual a 0,70 e 0,80. A Figura 6.16 indica que há três possíveis regiões de localização da face, sendo que as duas regiões das extremidades são falsos positivos (*outliers*), pois não correspondem à região real da face, como esperado. A Figura 6.17 apresenta apenas a região real onde está localizada a face.

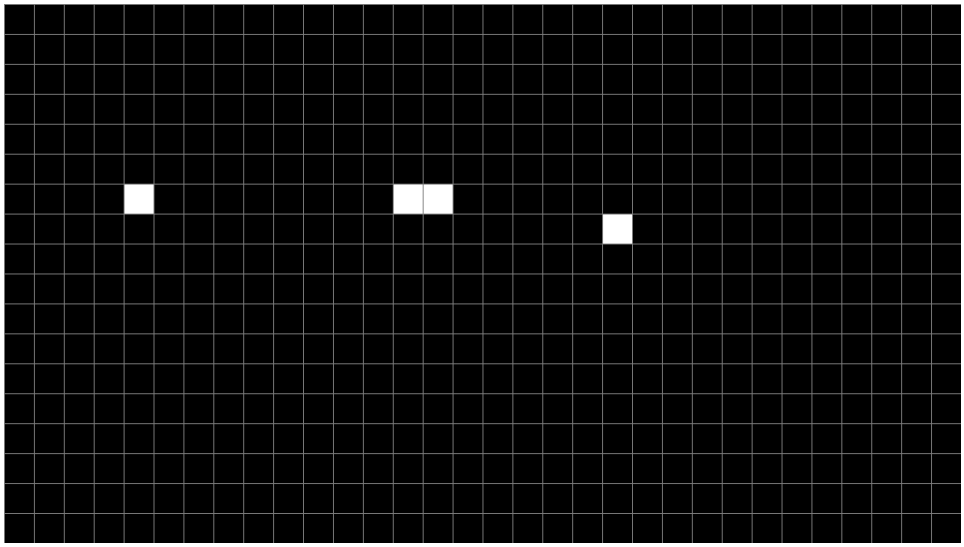

Figura 6.16: O processamento do algoritmo Blocos Particionados, na imagem Alcatraz, tendo a face como região modelo, determinou três possíveis regiões de localização da face do interlocutor. Utilizando como THRESHOLD = 0,70.

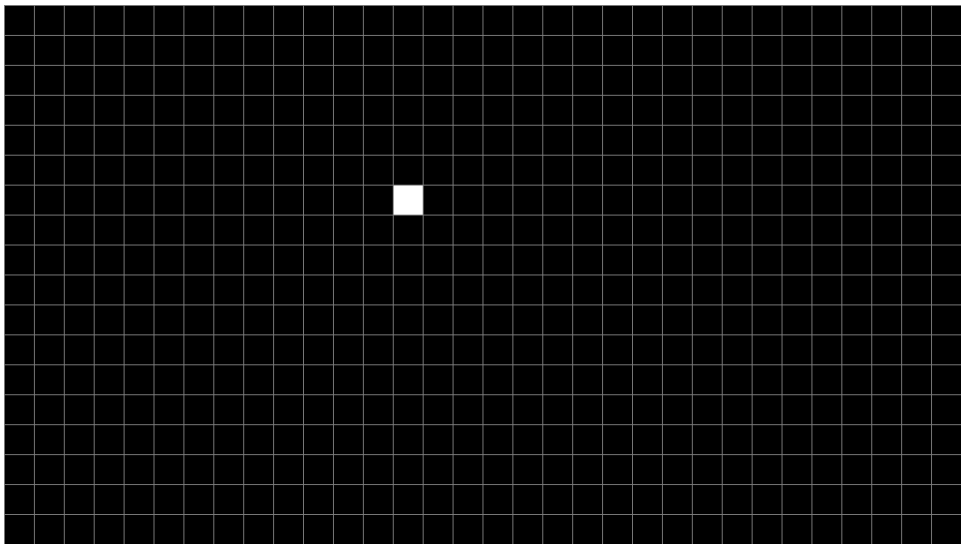

Figura 6.17: O processamento do algoritmo Blocos Particionados, na imagem Alcatraz, tendo a face como região modelo, determinou uma possível região de localização da face do interlocutor. Utilizando como THRESHOLD = 0,80.

A ocorrência de *outliers* deve-se à natureza da implementação de um algoritmo de processamento em tempo real. Como o algoritmo de deteção das áreas da face utiliza o histograma de cor do contorno dos blocos, quando a quantidade de detalhes do plano de fundo for bastante complexa há a possibilidade de serem definidas regiões que não representam o posicionamento da face do interlocutor. Assim, o processamento do *Particle Filter* considerará que essas regiões estão fora da localização da face, devido aos seus pesos insignificantes. Os pesos serão calculados na determinação do *importance sampling*, através do algoritmos de Módulo de cor e Módulo Gradiente. Esses calcularão pesos próximos de zero caso não encontrem, nesta região selecionada, as evidências da face e do seu contorno. A robustez deste método recai na exigência desta condição.

A Equação (6.1) apresenta a matriz com as distâncias calculadas para cada bloco da Figura 6.15, utilizando a técnica Blocos Particionados. Mesmo em uma imagem com um fundo repleto de diferentes detalhes, que dificultam a localização da região da face, é possível identificar os blocos com maior semelhança com a região modelo computando as distâncias de cada bloco.

$$\begin{bmatrix}
0.58 & 0.56 & 0.50 & 0.29 & 0.08 & 0.09 & 0.42 & 0.51 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.59 & 0.62 & 0.59 & 0.43 & 0.09 & 0.09 & 0.21 & 0.46 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.48 & 0.63 & 0.63 & 0.49 & 0.11 & 0.08 & 0.07 & 0.45 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.36 & 0.56 & 0.62 & 0.62 & 0.47 & 0.20 & 0.09 & 0.47 & 0.22 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.15 & 0.31 & 0.55 & 0.64 & 0.52 & 0.18 & 0.06 & 0.49 & 0.42 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.31 & 0.31 & 0.43 & 0.63 & 0.64 & 0.51 & 0.26 & 0.48 & 0.51 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.40 & 0.32 & 0.34 & 0.58 & 0.64 & 0.60 & 0.17 & 0.43 & 0.39 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.37 & 0.13 & 0.28 & 0.43 & 0.56 & 0.64 & 0.41 & 0.21 & 0.35 & 0.05 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.40 & 0.42 & 0.33 & 0.42 & 0.46 & 0.61 & 0.53 & 0.25 & 0.47 & 0.43 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.26 & 0.38 & 0.36 & 0.13 & 0.39 & 0.53 & 0.61 & 0.36 & 0.16 & 0.38 & 0.06 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.53 & 0.47 & 0.37 & 0.41 & 0.38 & 0.45 & 0.60 & 0.47 & 0.13 & 0.29 & 0.40 & 0.50 & 0.55 & 0.47 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.61 & 0.62 & 0.54 & 0.42 & 0.38 & 0.51 & 0.57 & 0.58 & 0.37 & 0.50 & 0.70 & 0.70 & 0.61 & 0.61 & 0.33 & 0.00 & 0.00 & 0.00 \\
0.58 & 0.60 & 0.62 & 0.57 & 0.44 & 0.38 & 0.50 & 0.57 & 0.49 & 0.39 & 0.48 & 0.54 & 0.61 & 0.60 & 0.25 & 0.06 & 0.00 & 0.00 \\
0.54 & 0.58 & 0.55 & 0.59 & 0.58 & 0.53 & 0.44 & 0.50 & 0.59 & 0.30 & 0.49 & 0.61 & 0.59 & 0.54 & 0.18 & 0.05 & 0.00 & 0.00 \\
0.55 & 0.54 & 0.58 & 0.56 & 0.57 & 0.57 & 0.56 & 0.54 & 0.53 & 0.46 & 0.37 & 0.49 & 0.58 & 0.56 & 0.27 & 0.00 & 0.00 & 0.00 \\
0.57 & 0.59 & 0.56 & 0.56 & 0.57 & 0.57 & 0.64 & 0.59 & 0.57 & 0.52 & 0.42 & 0.57 & 0.61 & 0.52 & 0.35 & 0.04 & 0.00 & 0.00 \\
0.53 & 0.56 & 0.62 & 0.55 & 0.53 & 0.61 & 0.62 & 0.47 & 0.53 & 0.62 & 0.50 & 0.60 & 0.54 & 0.54 & 0.34 & 0.00 & 0.00 & 0.00 \\
0.57 & 0.52 & 0.52 & 0.55 & 0.45 & 0.46 & 0.55 & 0.37 & 0.33 & 0.34 & 0.52 & 0.75 & 0.54 & 0.31 & 0.30 & 0.00 & 0.00 & 0.00 \\
0.50 & 0.50 & 0.38 & 0.50 & 0.53 & 0.43 & 0.35 & 0.38 & 0.24 & 0.37 & 0.62 & 0.84 & 0.63 & 0.19 & 0.13 & 0.00 & 0.00 & 0.00 \\
0.50 & 0.54 & 0.60 & 0.60 & 0.53 & 0.54 & 0.59 & 0.47 & 0.42 & 0.50 & 0.58 & 0.67 & 0.47 & 0.06 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.56 & 0.61 & 0.64 & 0.63 & 0.64 & 0.64 & 0.62 & 0.54 & 0.51 & 0.58 & 0.57 & 0.46 & 0.14 & 0.36 & 0.12 & 0.00 & 0.00 & 0.00 \\
0.63 & 0.64 & 0.65 & 0.63 & 0.62 & 0.63 & 0.63 & 0.56 & 0.58 & 0.58 & 0.53 & 0.29 & 0.32 & 0.23 & 0.15 & 0.36 & 0.08 & 0.00 \\
0.65 & 0.63 & 0.62 & 0.63 & 0.61 & 0.63 & 0.63 & 0.55 & 0.64 & 0.67 & 0.62 & 0.39 & 0.32 & 0.14 & 0.39 & 0.49 & 0.45 & 0.45 \\
0.56 & 0.61 & 0.61 & 0.64 & 0.62 & 0.64 & 0.62 & 0.64 & 0.57 & 0.63 & 0.64 & 0.49 & 0.18 & 0.29 & 0.54 & 0.61 & 0.63 & 0.60 \\
0.61 & 0.60 & 0.60 & 0.61 & 0.65 & 0.62 & 0.65 & 0.61 & 0.62 & 0.57 & 0.64 & 0.58 & 0.34 & 0.41 & 0.52 & 0.59 & 0.62 & 0.60 \\
0.55 & 0.57 & 0.64 & 0.65 & 0.63 & 0.61 & 0.57 & 0.57 & 0.61 & 0.62 & 0.64 & 0.67 & 0.49 & 0.43 & 0.47 & 0.51 & 0.62 & 0.62 \\
0.60 & 0.57 & 0.64 & 0.63 & 0.62 & 0.55 & 0.55 & 0.60 & 0.58 & 0.61 & 0.63 & 0.66 & 0.52 & 0.28 & 0.40 & 0.51 & 0.62 & 0.52 \\
0.61 & 0.61 & 0.59 & 0.57 & 0.59 & 0.55 & 0.53 & 0.62 & 0.63 & 0.63 & 0.63 & 0.70 & 0.52 & 0.14 & 0.14 & 0.30 & 0.44 & 0.38 \\
0.63 & 0.57 & 0.57 & 0.56 & 0.57 & 0.55 & 0.58 & 0.63 & 0.64 & 0.67 & 0.62 & 0.52 & 0.50 & 0.33 & 0.32 & 0.21 & 0.30 & 0.28 \\
0.60 & 0.57 & 0.57 & 0.57 & 0.55 & 0.55 & 0.57 & 0.63 & 0.59 & 0.63 & 0.51 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.62 & 0.56 & 0.57 & 0.55 & 0.55 & 0.55 & 0.60 & 0.64 & 0.55 & 0.53 & 0.50 & 0.22 & 0.20 & 0.16 & 0.06 & 0.05 & 0.00 & 0.00 \\
0.57 & 0.56 & 0.57 & 0.57 & 0.55 & 0.55 & 0.59 & 0.64 & 0.49 & 0.47 & 0.41 & 0.11 & 0.14 & 0.15 & 0.07 & 0.10 & 0.09 & 0.00
\end{bmatrix} \quad (6.1)$$

6.5 Amostras de teste

Para testar a estabilidade e o desempenho do acompanhamento da face é necessário, primeiramente, desenvolver uma sequência instável de vídeo de teste, onde são conhecidos os parâmetros da deslocamento tanto no eixo horizontal quanto no eixo vertical, simulando, desta forma, uma câmera com uma variação indesejada no seu plano de visualização. Os parâmetros utilizados para todas as sequências construídas estão apresentados na Tabela 6.3. Cada sequência de vídeo possui 1000 quadros, que se deslocam no eixo horizontal e vertical aleatoriamente, seguindo uma distribuição Gaussiana que é descrita por esses parâmetros.

Tabela 6.3: Parâmetros das amostras de teste.

Eixo	Média	Variância
x	0	1
y	0	1
x	0	10
y	0	1
x	0	10
y	0	10

A partir dos vídeos gerados será aplicado o algoritmo *Particle Filter* para acompanhamento da face e assim será possível gerar um conjunto de resultados experimentais que fornecerão a comprovação necessária da eficácia deste método proposto

O primeiro vídeo de teste, com os parâmetros apresentado na Tabela 6.3, gera um conjunto de amostras de deslocamento em que as primeiras cem amostras são visualizadas na Figura 6.18. Esses valores representam o deslocamento tanto no eixo X como no eixo Y da imagem estática utilizada para gerar o vídeo. Verificando essas amostras geradas, é possível construir os seus respectivos histogramas do módulo do deslocamento em função do eixo horizontal e do eixo vertical, projetando a distribuição de probabilidades das duas amostras em ambas as direções, tal como é apresentado nas Figuras 6.19 e 6.20.

Para uma melhor visualização da distribuição de probabilidade do deslocamento no eixo Y, com alteração da escala do eixo horizontal (*bins*), a Figura 6.21 apresenta a mesma distribuição previamente apresentada na Figura 6.20, porém utilizando uma escala

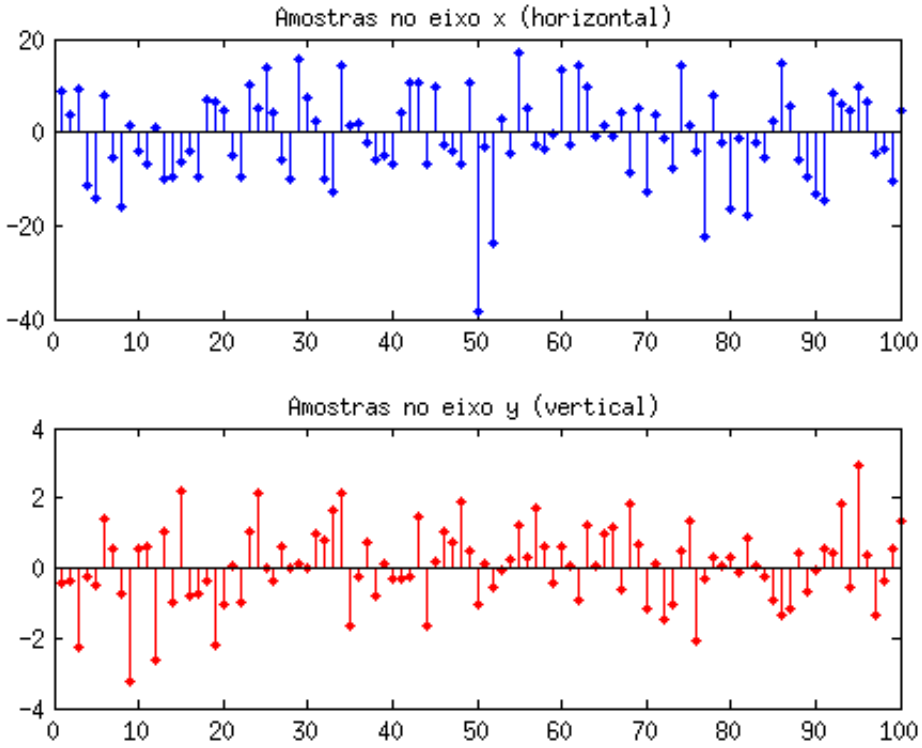

Figura 6.18: Amostras de teste com os parâmetros responsáveis em gerar os vídeos que serão utilizados para comprovar o funcionamento do algoritmo *Particle Filter* proposto neste trabalho. Os deslocamentos simulam uma instabilidade indesejada no vídeo que será compensada durante o processamento da filtragem.

para os *bins* mais adequada para visualização.

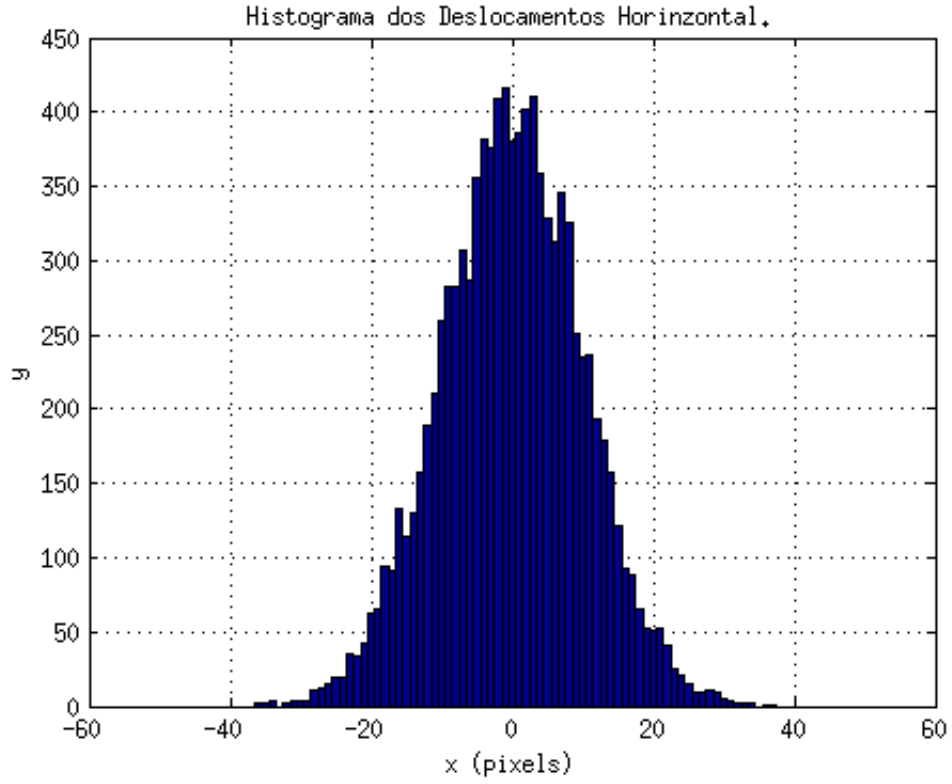

Figura 6.19: Amostras utilizadas para representar o deslocamento no eixo X.

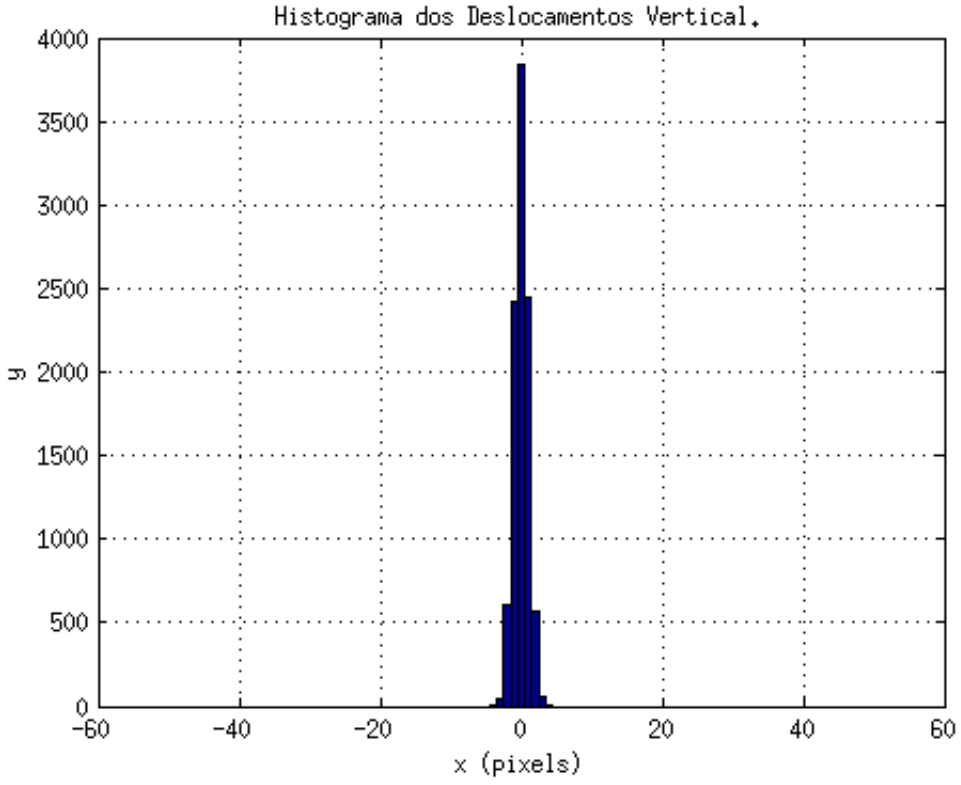

Figura 6.20: Amostras utilizadas para representar o deslocamento no eixo Y.

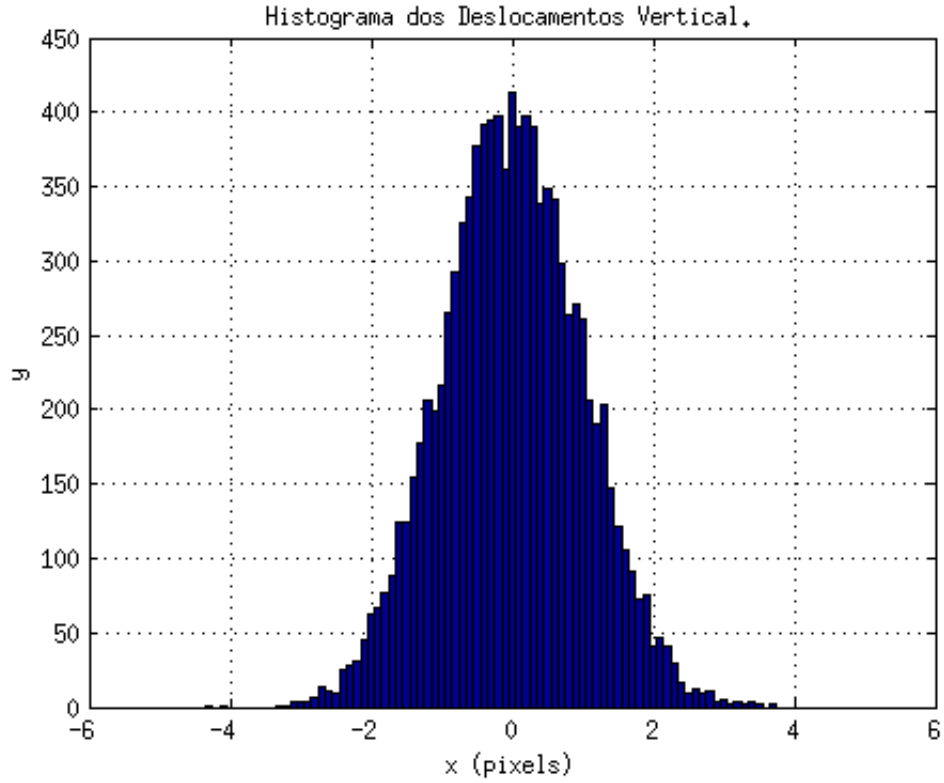

Figura 6.21: Amostras utilizadas para representar o deslocamento no eixo Y.

6.6 Vídeo Claire modificado

Utilizando as amostras de teste apresentadas na Seção 6.5, foi construído um vídeo contendo um plano de fundo (*background*) estático e utilizando um quadro da sequência de vídeo "Claire" que desliza sob este plano, simulando um vídeo com uma trepidação indesejável, como se fosse gerado a partir de uma câmera não estabilizada. O nome do vídeo é "Claire Deslizante". A Figura 6.22 apresenta nove quadros desse vídeo, obtidos aleatoriamente. O deslocamento nos eixos vertical e horizontal da imagem "Claire" são gerados a partir da amostra de teste com os parâmetros apresentados na Tabela 6.3.

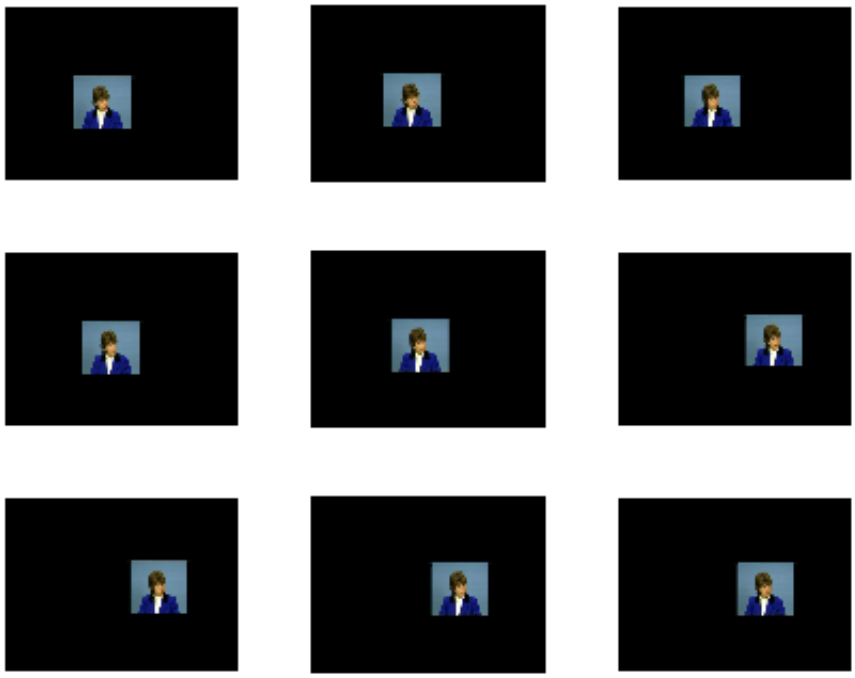

Figura 6.22: Amostras de quadros da sequência de vídeo Claire Deslizante.

A Figura 6.23 apresenta as distâncias não normalizadas, calculadas de cada bloco quadro de número sete do vídeo "Claire Deslizante" em relação a região selecionada da face da Claire. A partir desta informação o método Blocos Particionados identifica qual região deve ser considerada como a nova posição da face a ser utilizada pelo *Importance Function* para calcular as amostras do *Particle Filter*. A Figura 6.24 apresenta a mesma informação descrita pela Figura 6.23 porém com uma projeção em tons de cinza dos valores calculados para cada bloco.

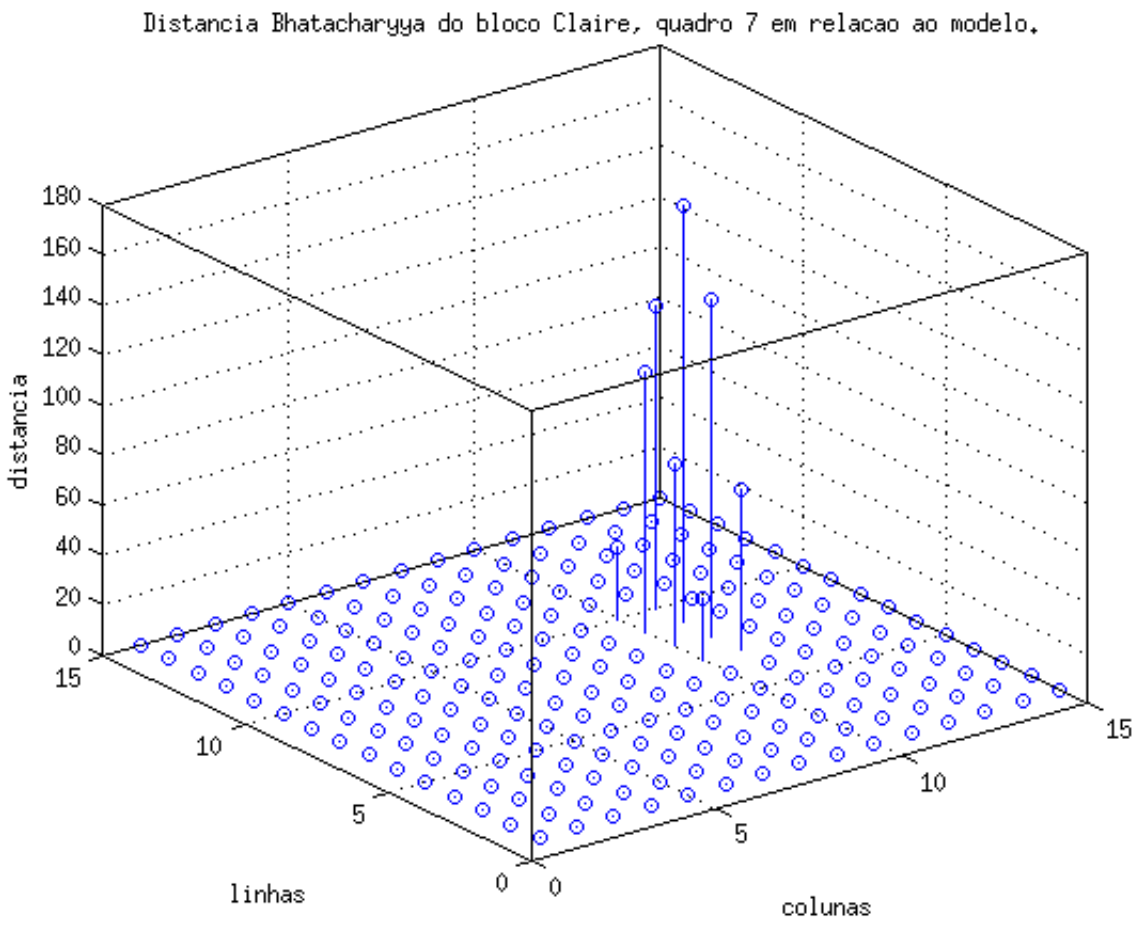

Figura 6.23: Amostras de quadros da sequência de vídeo Claire Deslizante.

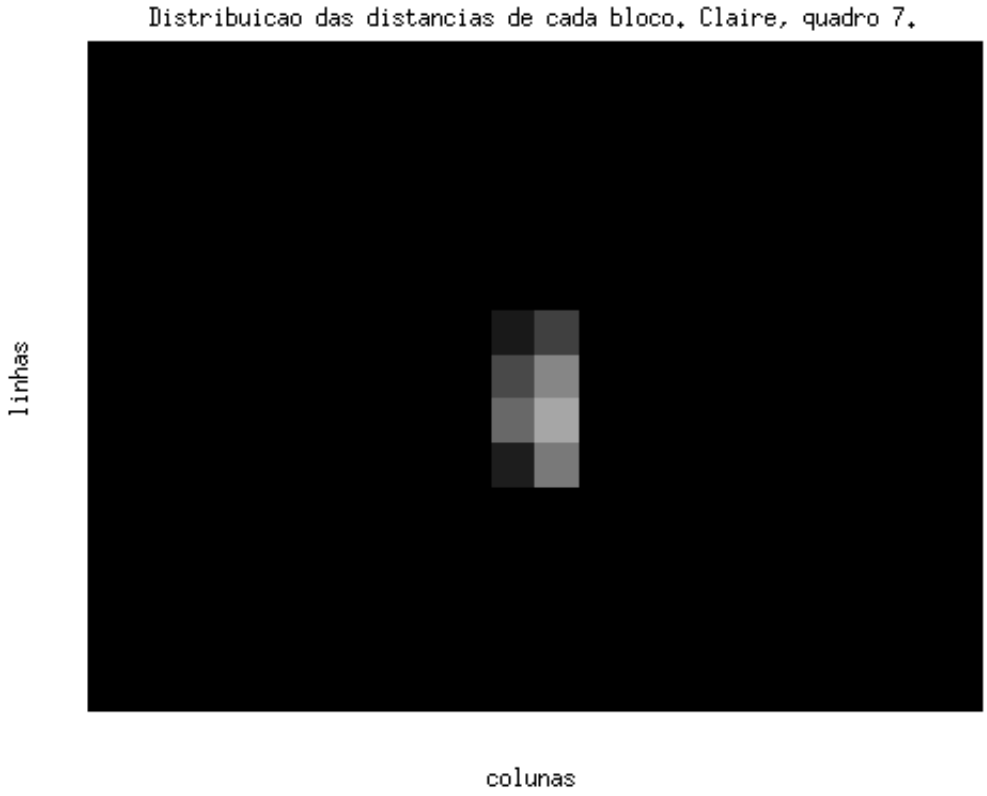

Figura 6.24: Amostras de quadros da sequência de vídeo Claire Deslizante rebatidos em cada bloco do quadro indicando em tons de cinza quais blocos tem maior probabilidade de conter a região da face esperada.

6.7 Vídeo Alcatraz

O vídeo intitulado "Alcatraz" é outra ferramenta para avaliar a eficiência do filtro proposto nesta dissertação. Ele foi gerado a partir de uma imagem estática, apresentada na Figura D.2, que se desloca quadro a quadro no plano X-Y, gerando uma sequência de imagens que simulam um vídeo contendo uma variação indesejável, porém conhecida. Os deslocamentos ocorrem aleatoriamente, com uma distribuição Gaussiana, em ambos os eixos, sendo que a translação no eixo horizontal possui sua variância definida em dez *pixels* ($\sigma_x = 10$), enquanto que a translação no eixo vertical possui sua variância definida em um *pixel* ($\sigma_y = 1$). Com essa sequência de teste, é possível verificar com precisão a diferença entre os valores reais de deslocamento do vídeo e os valores estimados, utilizando a técnica de *Particle Filter* desenvolvida no Capítulo 4.

A Figura 6.25 apresenta uma sequência de quatro quadros do vídeo de teste, obtidos de modo a facilitar a visualização da diferença de deslocamento horizontal e vertical entre os quadros. A Figura 6.26 mostra a localização do centro da elipse que representa a face do interlocutor nas vinte e cinco primeiras amostras aplicadas ao vídeo, tanto no eixo horizontal quanto no eixo vertical. A Figura 6.27 apresenta a diferença inter-quadros das mesmas amostras apresentada na Figura 6.26, a fim de possibilitar uma melhor visualização da translação quadro a quadro gerada neste vídeo.

Para demonstrar a utilização e a eficiência do *Particle Filter*, é utilizado o cenário descrito na Tabela 6.4.

O cenário de teste configura o processador do *Particle Filter* com cinquenta partículas, $N_p = 50$, e a matriz de variância definida por

$$\begin{bmatrix} \sigma_{x_c} \\ \sigma_{y_c} \\ \sigma_b \\ \sigma_\phi \end{bmatrix} = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix} , \tag{6.2}$$

que identifica a distribuição de probabilidades utilizada pelo *importance sampling* para gerar as amostras aleatórias associadas a cada partícula $\mathbf{X}_k^i$.

A Figura 6.28 apresenta a elipse parametrizada descrevendo o estado da face utilizado como modelo de comparação com cada partícula gerada. As partículas geradas na primeira iteração do filtro, em $k = 0$ são representadas na Figura 6.29.

Figura 6.25: Sequência dos quadros 1, 2, 4 e 15, do vídeo Alcatraz.avi.

O processamento do filtro considera o instante $k = 0$ como momento quando é selecionado a região modelo da face, dando inicio ao acompanhamento. O *Particle Filter* inicia a geração do conjunto de partículas, estipulando para cada elemento o mesmo peso inicial $\{\mathbf{X}_{k=0}^{i}\}_{i=1}^{N_p} = \frac{1}{N_p}$. A Figura 6.30 ilustra a distribuição das partículas projetadas no eixo horizontal.

A Figura 6.31 compara os resultados obtidos do acompanhamento da face. Os valores em azul representam o deslocamento no eixo XY real do centro da face, enquanto que os valores em vermelho representam os valores estimados pelo processamento do *Particle Filter*. A Figura 6.32 apresenta a diferença, ou erro, entre os valores reais e estimados desta mesma sequência.

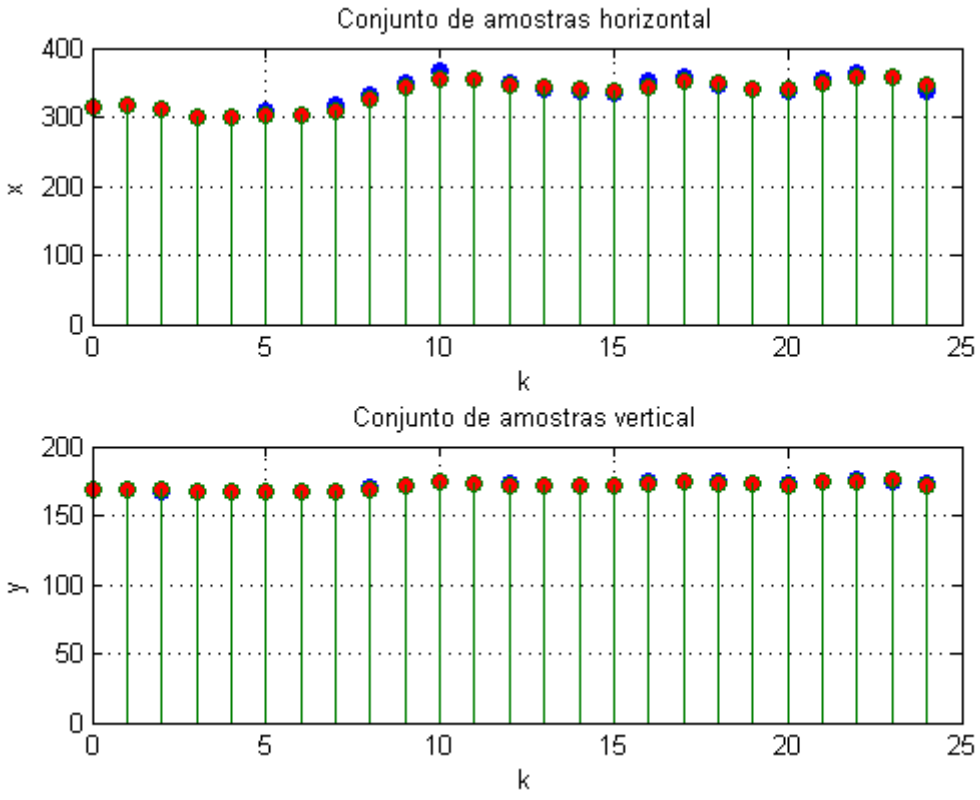

Figura 6.26: Sequência dos quadros 1, 2, 4 e 15, do vídeo Alcatraz.avi.

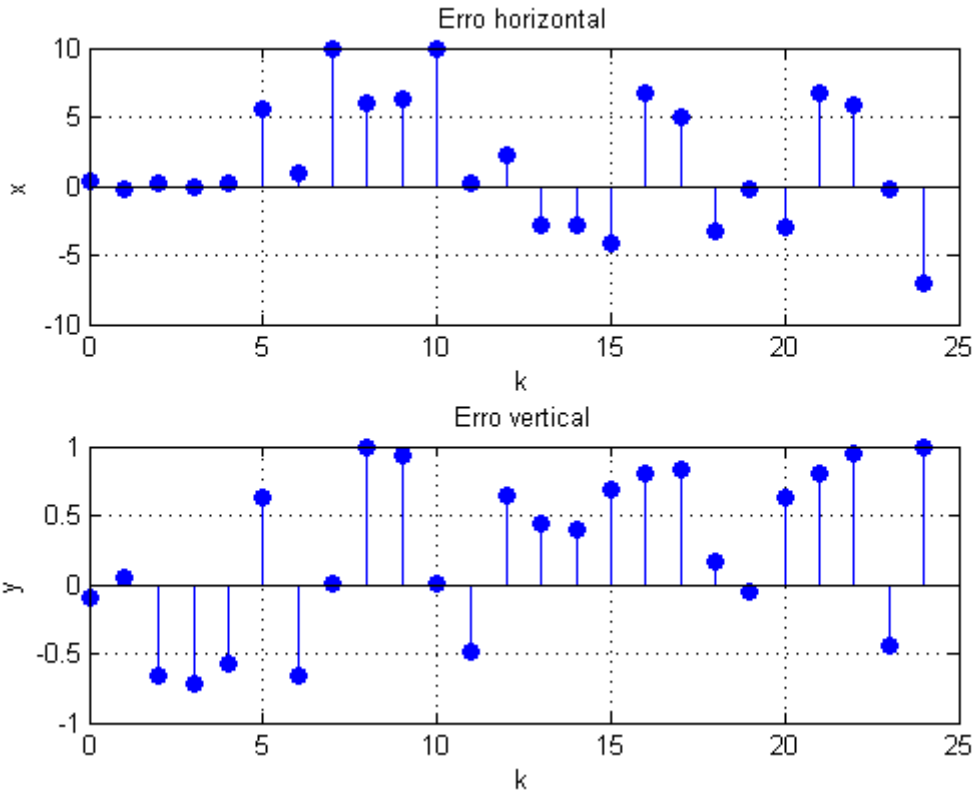

Figura 6.27: Diferença inter-quadros das primeiros vinte e cinco amostras do vídeo Alcatraz.avi.

Tabela 6.4: Parâmetros utilizados para gerar as partículas do filtro aplicado ao vídeo Alcatraz.

Número de partículas	σ_x	σ_y	σ_b	σ_ϕ
50	10	5	5	5

Figura 6.28: Representação da elipse parametrizada, definida como modelo do estado da face, no vídeo Alcatraz.avi, a ser acompanhado pelo algoritmo de *Particle Filter*.

A Figura 6.33 apresenta um comparativo entre o tempo de processamento para cada quadro realizado pelo algoritmo Blocos Particionados para encontrar o deslocamento da face do interlocutor quadro-a-quadro. Esse comparativo demonstra o ganho no tempo de processamento total num dado instante de tempo k quando são utilizadas somente os *pixels* da borda de cada bloco em relação ao mesmo processamento, mas utilizando todos os *pixels* contidos no mesmo quadro. A razão do ganho médio foi de aproximadamente 3,8 vezes quando utilizado somente as bordas como sugere o algoritmo.

Figura 6.29: Representação gráfica das cinquenta partículas aleatórias, geradas no quadro $k = 0$, do vídeo Alcatraz.avi, a partir de uma distribuição Gaussiana de média zero e matriz de variâncias definida pela Equação (6.2). As elipses em cor azul representam as partículas geradas para definir a posição estimada da face, representada pela elipse de cor vermelha.

Figura 6.30: Representação gráfica dos pesos das cinquenta partículas aleatoriamente geradas, no quadro $k = 0$, do vídeo Alcatraz.avi, a partir de uma distribuição Gaussiana de média zero e matriz de variâncias definida pela Equação (6.2), projetadas no eixo horizontal, indicando a localização da elipse parametrizada que modela o estado da face.

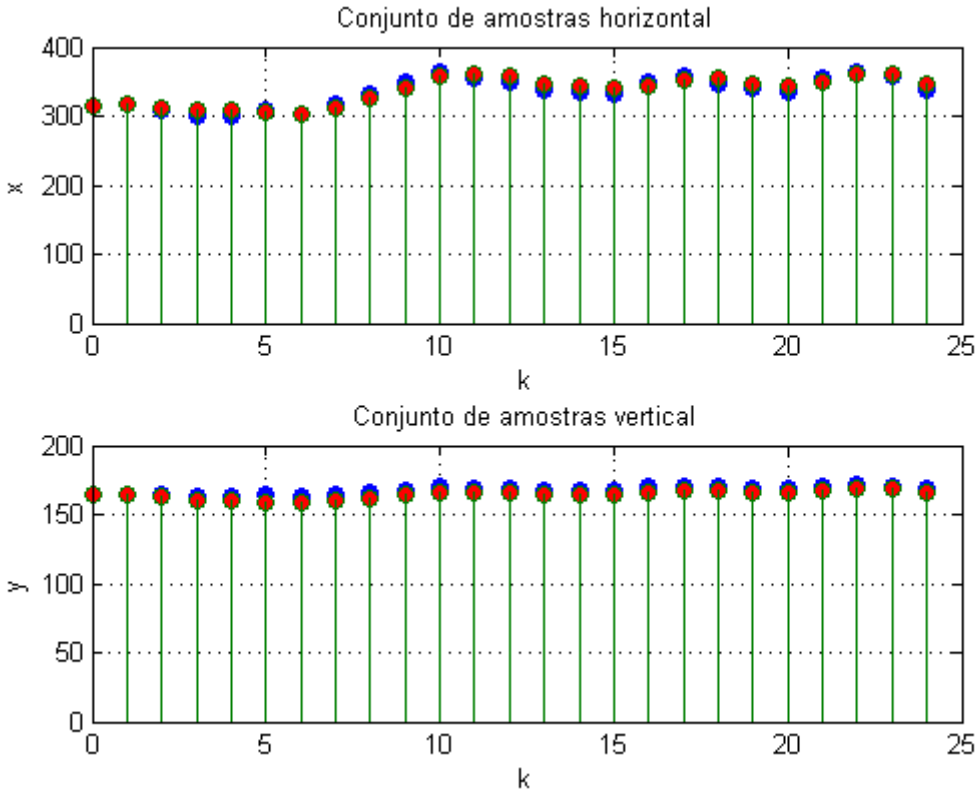

Figura 6.31: Comparação entre os vinte e cinco primeiros estados estimados da face, gerados pelo algoritmo *Particle Filter*, e os valores reais para os estados previamente definidos como amostras de teste. Os valores na cor azul representam a projeção do centro da elipse no eixo horizontal x_c como também no eixo vertical y_c do vetor de estados reais $\mathbf{x}_k$. Os valores na cor vermelho representam as estimativas das projeções do vetor de estados $\mathbf{X}_k$.

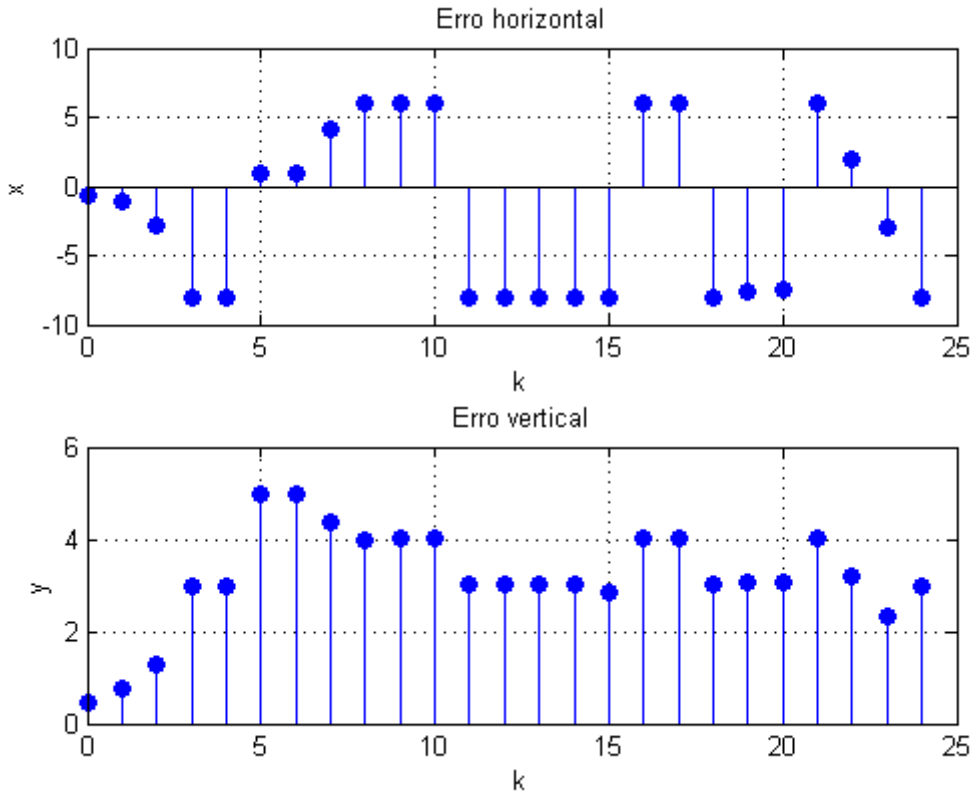

Figura 6.32: Comparação do erro de estimativa entre os vinte e cinco primeiros estados estimados da face, gerados pelo algoritmo *particle filter*, e os valores reais para os estados previamente definidos como amostras de teste. Os valores na cor azul representam a projeção do centro da elipse no eixo horizontal x_c como também no eixo vertical y_c do vetor de estados reais $\mathbf{x}_k$. Os valores na cor vermelho representam as estimativas das projeções do vetor de estados $\mathbf{X}_k$.

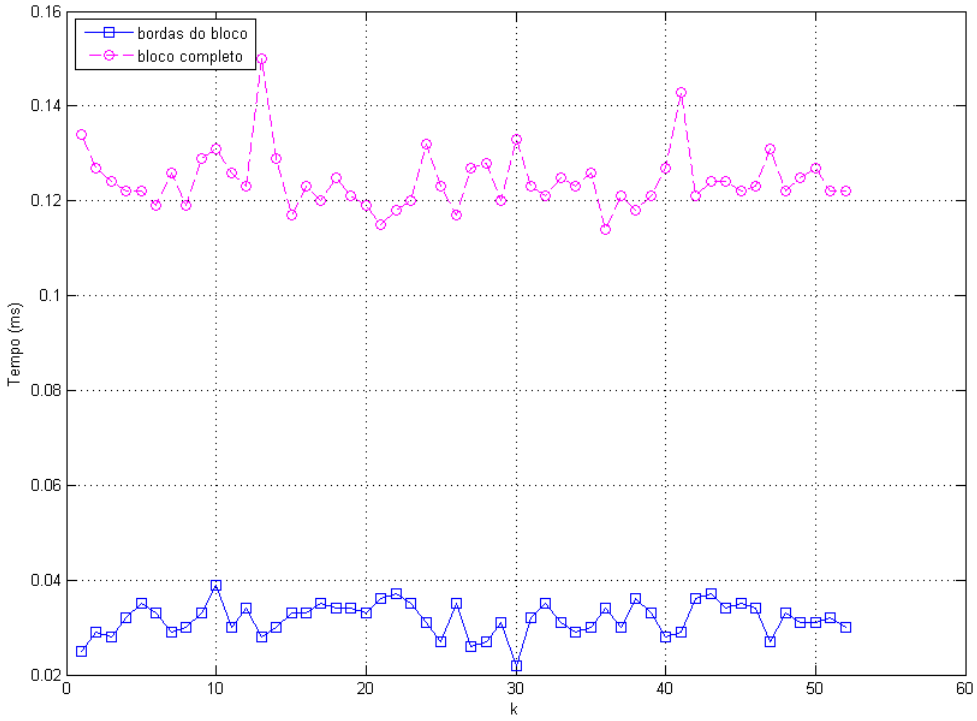

Figura 6.33: Comparação quadro-a-quadro entre o tempo de processamento do algoritmo Blocos Particionados. O histograma de cor de todos os blocos de 32×32 *pixels* do quadro utiliza um tempo processamento médio de $0,123$ ms enquanto que o cômputo utilizando somente as bordas de cada bloco possui um processamento médio de $0,032$ ms. A razão entre as respectivas médias de tempo foi de aproximadamente 3,8.

Capítulo 7

Conclusões e trabalhos futuros

Este trabalho apresentou os fundamentos da técnica denominada de *Particle Filter*, introduzindo conceitos importantes para o entendimento do conteúdo apresentado. A metodologia abordada foi empregada nos primeiros trabalhos de computação visual e recebeu bastante atenção e popularidade pelo fato de conseguir resolver diretamente problemas de estimação em sistema dinâmicos com distribuição de probabilidade *a posteriori* não linear e não gaussiano. Até então, muitos trabalhos científicos procuravam resolver este tipo de cenário com adaptações em sistemas lineares, tal como a filtragem por Kalman.

Uma contribuição que este trabalho realizou foi estender o conceito clássico da filtragem por *particle filter*, utilizando não apenas a informação *a priori* da dinâmica do movimento da face, mas também a informação referente a amostra corrente $\mathbf{z}_k$, através da indicação do vetor de deslocamento da face $\Delta\mathbf{X}_k$ para gerar a *importance function*. Esta abordagem tem inúmeras vantagens em relação à abordagem clássica. Ela aumenta a eficácia do processamento do *particle filter*, diminuindo o número total de amostras aleatórias. Permite o processamento do filtro em toda a extensão do quadro. Habilita o algoritmo para trabalhar em ambiente computacional do tipo multitarefa. Ainda, com pouca modificação do algoritmo original, possibilita a generalização do conceito de acompanhamento da face para qualquer tipo de objeto, desde que definido um modelo para o vetor de estados adequado.

No Capítulo 4, foi introduzida a técnica batizada como "Blocos Particionados" (BP). Ela mostrou-se eficiente na busca das regiões da face no quadro $\mathbf{z}_k$, através da comparação do histograma de cor de um número pequeno de *pixels* pertencentes a uma determinada área da imagem, definida por um bloco, com o histograma de referência pre-

viamente definido e que descreve a região selecionada da face do interlocutor, definindo, consequentemente, o vetor de deslocamento inter-quadros da face do interlocutor. A facilidade de uso desta técnica e sua rapidez na definição das regiões possíveis de localizar a face permite utilizá-lo em toda a extensão do quadro $\mathbf{z}_k$. Ele mostrou-se promissor na aplicação em vídeo de alta definição, pois o seu custo de processamento não aumenta significativamente com o aumento das dimensões dos quadros de vídeo. Esta técnica supera os requisitos utilizados na técnica de *Adaptive Block Matching*, que considera apenas uma região limitada do quadro $\mathbf{z}_k$ para a busca do vetor de deslocamento $\Delta\mathbf{X}_k$, além do processamento intenso na busca dos blocos pertencentes à região de busca, substituido por uma comparação de um histograma de cor, construído a partir de uma baixa dimensão (*bins*).

A implementação utilizada para investigar o algoritmo BP conseguiu encontrar com rapidez e robustez a face do interlocutor, mesmo em cenários com o plano de fundo bem heterogêneo como a imagem "Alcatraz".

Dando continuidade a esse trabalho em futuras pesquisas, há temas que devem ser aperfeiçoados:

- O primeiro está relacionado à seleção da área da face utilizada como referência. Este trabalho utilizou um modelo invariante no tempo. Uma melhoria seria criar um técnica adaptativa que varie o estado do modelo de acordo com as possíveis variações do ambiente da cena, tal como iluminação externa e ganho absoluto dos *pixels*, muito comum em câmeras que possuem processamento de imagem embarcado, tal como as câmeras USB utilizadas em computadores pessoais.

- A técnica "Blocos Particionados" utilizou apenas um modelo de escolha dos *pixels* da borda de cada bloco. A Figura 7.1 mostra outras possíveis abordagens para a escolha dos *pixels*, interna ao bloco, que poderiam ser investigadas.

- Incorporar à técnica "Blocos Particionados" o algoritmo *Adaptive Block Matching* (ABM) [BS04] somente nas regiões selecionadas pelo BP. Isso aperfeiçoa o cálculo do vetor de deslocamento $\Delta\mathbf{X}_k$ da face, pois utiliza as vantagens de ambas as técnicas num só processamento. Do lado do BP utiliza a vantagem de conseguir processar toda a extensão do quadro $\mathbf{z}_k$ com baixo custo, enquanto que do lado do ABM, consegue-se uma melhor precisão no cálculo do vetor de deslocamento, mas diminu-

indo consideravelmente seu custo, pois somente as regiões definidas pelo BP serão processadas, ao invés de toda uma janela pré-definida.

- Desenvolver uma análise quantitativa da melhoria que as técnicas de compactação de vídeo recebem ao preprocessar um vídeo que possui uma variação indesejada através do algoritmo de estabilização proposto. Um vídeo que possui uma estabilização global de sua imagem permite uma compressão superior se comparado a um vídeo que possui uma variação global do seu conteúdo. Nesse cenário, a técnica proposta de estabilização diminui a variação global permitindo aos algoritmos de compactação melhores taxas de compressão.

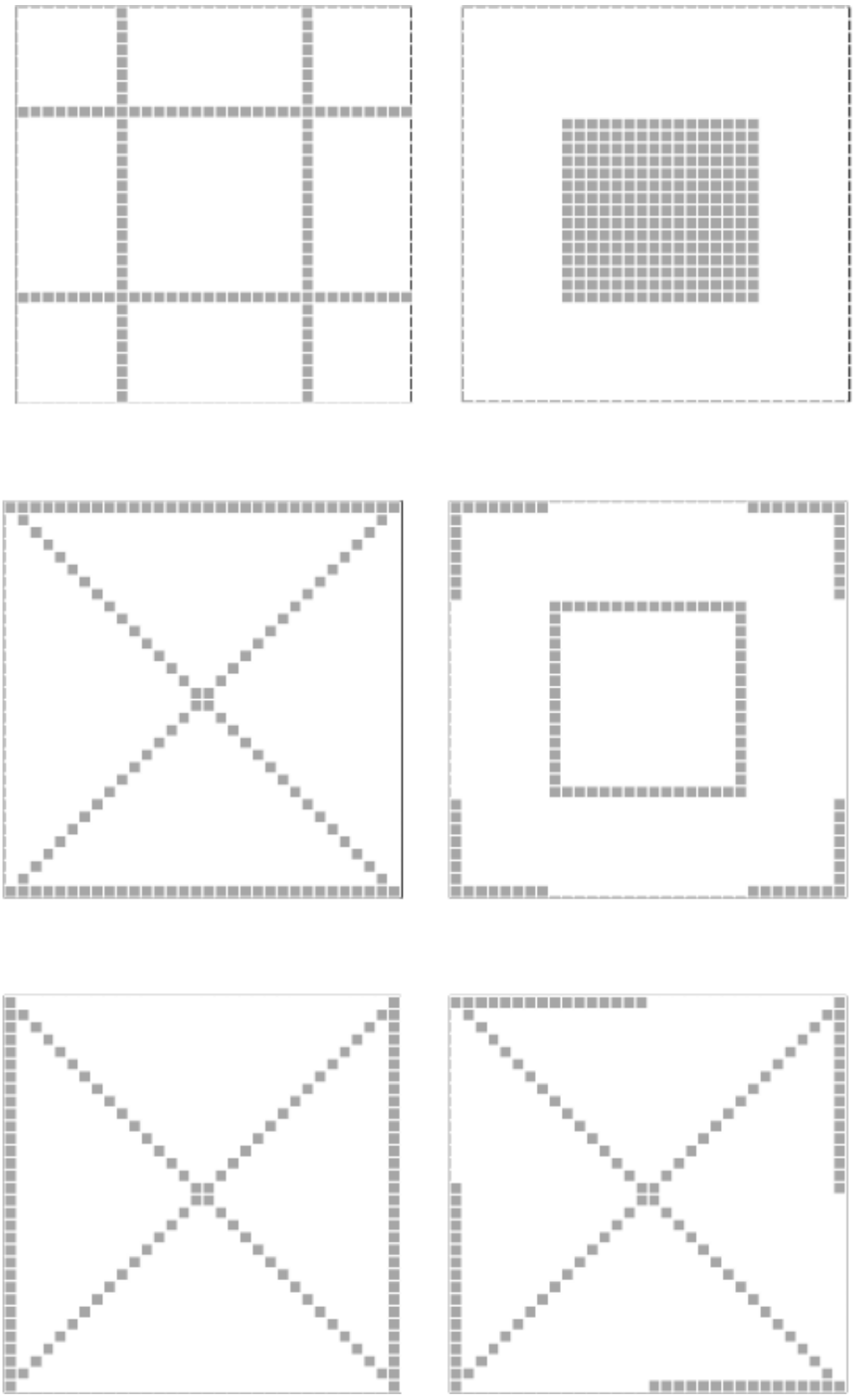

Figura 7.1: Outras maneiras de seleção dos *pixels*, interna ao bloco, para uso do BP que podem ser investigadas em trabalhos futuros.

Apêndice A

Notação utilizada

Toda apresentação matemática desta dissertação é realizada utilizando o sinais discretos no tempo. Portanto, qualquer referência temporal embute a noção de intervalos constantes no tempo. Primariamente, foi utilizado a variável k em subescrito para identificar qualquer índice no tempo, onde $k \in \mathbb{N}$.

Notação	**Significado**
$b_k^{i,j}$	bloco 32×32 com borda superior esquerda localizada no *pixel* (i, j) do quadro $\mathbf{z}_k$
$bb_k^{i,j}$	matriz contendo os *pixels* da borda $b_k^{i,j}$
$\mathbf{B}$	canal de cor azul do espaço de cor RGB
$B_{mse}(.)$	função de erro médio quadrático
$\mathbf{C}$	matriz de restrição de contorno da elipse
$\mathcal{D}_{\mathcal{A}}(a)$	soma quadrática das distâncias dos pontos a de uma supefície definida pelo espaço vetorial $\mathcal{A}$
$d_B(a, b)$	distância Bhattacharyya para comparação entre os histogramas a e b
$\delta\mathbf{x}_k$	operador diferença entre os vetores $\mathbf{x}_k$ e $\mathbf{x}_{k-1}$
$\mathbf{D}$	matriz de *design* com a definição do pontos (x, y) no plano da imagem $\Re^2$
$E\{.\}$	função valor esperado ou média
$\mathbf{f}_k(.)$	função de transição do sistema dinâmico no tempo k
$F(.)$	função da superfície cônica em $\Re^2$
$\mathbf{G}$	canal de cor verde do espaço de cor RGB

Notação	Significado
$\mathbf{h}_k(.)$	função de observação do sistema dinâmico no tempo k
I_k	integral pelo método Monte Carlo
$J(.)$	função soma quadrática
$H_{\mathcal{S}}(.)$	histograma de cor definido no sub-espaço vetorial $\mathcal{S}$
$hbb_k^{i,j}$	histograma da matriz $bb_k^{i,j}$
$\{k \mid k \geq 0\}$	inteiro representando o índice de tempo discreto
$\mathbf{L}$	matriz de Cholesky
$\mathbb{N}$	conjunto dos inteiros não negativos, $\{0, 1, 2, \cdots\}$
N_{eff}	*effective sample size*
$\mathcal{N}(\mu, \Sigma^2)$	processo gaussiano de média μ e variância Σ
N_p	número total de partículas do *Particle Filter*
N_r	número total de regiões definidas pelo *particle function*
N_T	limite de partículas do *Particle Filter* que define a necessidade de *resampling*
$\hat{N}_{eff}$	estimativa da *effective sample size*
ν_{borda}^i	módulo gradiente da i-ésima partícula (*particle*)
ν_{cor}^i	módulo de cor da i-ésima partícula (*particle*)
ν_k^i	*particle likelihood* da i-ésima partícula (*particle*)
$\pi_k(.)$	função de importancia (*importance function*)
$\psi_i(.)$	i-ésima função de restrição
$\mathbf{p}_k$	vetor de estado da janela deslizante $\mathbf{P}$ no instante de tempo k
$p(\mathbf{x}_{0:k})$	distribuição de probabilidade *a priori* do conjunto de estados $\mathbf{x}_{0:k}$
$p(\mathbf{x}_k \mid \mathbf{x}_{0:k-1})$	distribuição conjunta de probabilidade da transição de estados do sistema dinâmico
$p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$	distribuição marginal de probabilidade da transição de estados do sistema dinâmico
$p(\mathbf{x}_k, \mathbf{z}_k)$	distribuição de probabilidade conjunta entre $\mathbf{x}_k$ e $\mathbf{z}_k$ quando independentes entre si
$p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$	distribuição de probabilidade marginal *a posteriori*
$p(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k})$	distribuição de probabilidade conjunta *a posteriori*
$p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})$	distribuição de probabilidade conjunta *a posteriori* não normalizada

Notação	**Significado**
$p(\mathbf{z}_{1:k})$	distribuição de probabilidade da evidência, representado pelo conjunto de observações $\mathbf{z}_{1:k}$ do sistema
$p(\mathbf{z}_k \vert \mathbf{x}_k)$	distribuição de probabilidade marginal de verossimilhança
$p(\mathbf{z}_{1:k} \vert \mathbf{x}_{0:k})$	distribuição de probabilidade conjunta de verossimilhança
$q(.)$	distribuição de probabilidade proposta (*proposal distribution*)
$\mathbf{P}$	janela delizante
$\mathbf{R}$	canal de cor vermelho do espaço de cor RGB
$\mathbf{S}$	matriz Scatter
$\mathbf{T}$	matriz de translação
θ_i	i-ésimo parâmetro de estimação
$\mathbf{v}_k$	processo de ruído do sistema dinâmico no tempo k
$\mathbf{V}$	matriz latice de amostragem ortogonal de vídeo
$Var(.)$	função variância
$\varphi_i(.)$	i-ésima função base ou portadora
$\mathbf{x}_k$	vetor de estados que modela o sistema dinâmico
$\mathbf{x}_{0:k}$	conjunto de estados do sistema dinâmico desde o valor inicial $\mathbf{x}_0$ até o tempo k, $\{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k\}$
$\hat{\mathbf{x}}_k$	estimação do vetor de estados $\mathbf{x}_k$ no tempo k
$\mathbf{X}$	variável randômica do espaço de estados
$\mathcal{X}^n$	espaço vetorial n-dimensional do vetor de estados do sistema
$\mathbf{X}_k^i$	amostra do vetor de estados pertencente a i-ésima partícula (*particle*) do filtro
$\{\mathbf{X}_k^i, W_k^i\}$	i-ésimo par amostra/peso do conjunto de partículas (*particle*) do filtro
$\mathbf{y}_k$	saída real do sistema dinâmico sem ruído de amostragem
$\mathbf{z}_{1:k}$	conjunto de observações do sistema dinâmico até o tempo k, $\{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \cdots, \mathbf{z}_k\}$
$\mathbf{z}_k$	vetor de observação do sistema dinâmico discreto no tempo k
$\mathbb{Z}$	conjunto de todos os inteiros
Z_k	probabilidade total de uma função randômica
w_k^i	peso não normalizado do i-ésima partícula (*particle*)
W_k^i	peso normalizado do i-ésima partícula (*particle*) do filtro

Apêndice B

O processo gaussiano

Uma variável randômica Gaussiana representada pelo vetor $1 \times N$ υ, $\upsilon \in \Re^n$, é determinada por sua média μ_υ e covariância $\mathbf{\Sigma}_\upsilon$, tal que

$$\mu_\upsilon = E\{\upsilon\} \tag{B.1a}$$

$$\mathbf{\Sigma}_\upsilon = E\{(\upsilon - \mu)(\upsilon - \mu)'\} \tag{B.1b}$$

.

Se $\mathbf{\Sigma}_\upsilon$ é uma matriz não singular, ou seja, possui inversa Σ_υ^{-1}, a densidade de probabilidade é representada pelo vetor $1 \times N$

$$p_V(\upsilon) = \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}_\upsilon|^{1/2}} \exp\left[-\frac{1}{2}(\upsilon - \mu_\upsilon)\mathbf{\Sigma}_\upsilon^{-1}(\upsilon - \mu_\upsilon)'\right] \quad . \tag{B.2}$$

A notação $N_V(\mu_\upsilon, \mathbf{\Sigma}_\upsilon)$ é utilizada nesta dissertação para representar a densidade probabilidade de um processo Gaussiano como descrito pela Equação [B.2].

B.1 Geração de amostras aleatórias com distribuição Gaussiana

Como apresentado na Seção 4.4, é necessário gerar amostras com distribuição gaussiana multivariada. Isso requer a implementação de um algoritmo baseado na transformação Box-Muller [PTVF07], que utiliza um conjunto de amostras aleatória de distribuição constante para gerar um novo conjunto de amostras aleatória com distribuição gaussiana de

média e variância conhecida, com a distribuição de densidade de probabilidade definida conforme a Equação (B.2).

Para o caso em que o vetor υ possui dimensão unitária, $\upsilon \in \Re^1$, média zero e variância unitária $N_\upsilon(0,1)$, a Equação (B.2) transforma-se em

$$p_V(\upsilon) = \frac{1}{(2\pi)^{1/2}} \exp\left[-\frac{1}{2}\upsilon^2\right] \quad . \tag{B.3}$$

Utilizando a distribuição de Weibull [PTVF07], dado duas distribuições uniformes x_1 e x_2, com valores positivos menor ou igual a um $(0 < x_1 \leq 1, 0 < x_2 \leq 1 | x_1, x_2 \in \Re)$ podem ser criadas outras duas distribuições gaussianas independentes, y_1 e y_2, utilizando a transformação

$$y_1 = \sqrt{-2\ln x_1}\cos(2\pi\, x_2) \tag{B.4a}$$

$$y_2 = \sqrt{-2\ln x_1}\sin(2\pi\, x_2) \quad . \tag{B.4b}$$

Entretanto, essa transformação proposta em (B.4) torna-se numericamente instável quando $x_1 \approx 0$. Para superar esta limitação, é utilizado a forma polar da transformação de Box-Muller [PTVF07], dado pelo seguinte algoritmo, onde a função $rand()$ retorna um valor entre 0 e 1, com distribuição uniforme:

Deve-se notar que y_1 e y_2 são ortogonais entre si e geram um conjunto de valores com distribuição gaussiana de média zero e variância igual a um.

$$y_1 = N(0,1)$$

$$y_2 = N(0,1)$$

Para o caso geral, multivariado, como todos os parâmetros do vetor υ apresentado na Equação (B.2) são independentes entre si, pode-se considerar a matriz de covariância com a forma diagonal unitária $\mathbf{\Sigma}_\upsilon = \mathbf{I}$. Esta configuração permite considerar cada

elemento do vetor v como uma distribuição independente.

Algorithm B.1.1: TRANSFORMACAO BOX-MULLER($rand()$)

$init$:

$x1 \leftarrow 2.0 * rand() - 1.0$

$x2 \leftarrow 2.0 * rand() - 1.0$

$w \leftarrow x_1^2 * x_2^2$

if $w \geqslant 1.0$

then goto $init$:

$w = \sqrt{\frac{-2.0 * \ln w}{w}}$

$y_1 \leftarrow x_1 * w$

$y_2 \leftarrow x_2 * w$

Apêndice C

Demonstração algébrica de equações do texto

Visando a concisão textual, algumas demonstrações algébricas foram removidas do texto principal e passados para os tópicos a seguir.

C.1 Demonstração algébrica da Equação (2.18)

A densidade de probabilidade de verossimilhança $p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})$ pode ser decomposta nos termos referentes ao tempo k e ao conjunto de termos de 0 até k-1 [Can09], de modo que

$$p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k}) = p(\mathbf{z}_k, \mathbf{z}_{1:k-1}|\mathbf{x}_k, \mathbf{x}_{0:k-1}) \ .$$

Pela regra da cadeia [Pap91], temos $p(a,b) = p(a|b)\ p(b)$. Fazendo $a = \mathbf{z}_k$ e $b = (\mathbf{z}_{1:k-1}|\mathbf{x}_k, \mathbf{x}_{0:k-1})$, temos

$$p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k}) = p(\mathbf{z}_k|\mathbf{z}_{1:k-1}, \mathbf{x}_k, \mathbf{x}_{0:k-1})\ p(\mathbf{z}_{1:k-1}|\mathbf{x}_k, \mathbf{x}_{0:k-1}) \ .$$

Como $\mathbf{z}_k$ é independente de $\mathbf{x}_{0:k-1}$ e $\mathbf{z}_{1:k-1}$, o termo $p(\mathbf{z}_k|\mathbf{z}_{1:k-1}, \mathbf{x}_k, \mathbf{x}_{0:k-1})$ torna-se $p(\mathbf{z}_k|\mathbf{x}_k)$ e o termo $p(\mathbf{z}_{1:k-1}|\mathbf{x}_k, \mathbf{x}_{0:k-1})$ torna-se $p(\mathbf{z}_{1:k-1}|\mathbf{x}_{0:k-1})$.

Assim a expressão final é representada pela Equação (2.18), a qual é repetida abaixo por conveniência.

$$p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k}) = p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{z}_{1:k-1}|\mathbf{x}_{0:k-1})$$

Apêndice D

Imagens de teste

Figura D.1: Imagem Lena.

Figura D.2: Imagem Alcatraz.

Figura D.3: Quadro da sequência de vídeo Claire.

Referências Bibliográficas

[AMGC02] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[Bir98] Stan Birchfield. Elliptical Head Tracking Using Intensity Gradients and Color Histograms. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, page 232, 1998.

[Boo79] F. L. Bookstein. Fitting Conic Sections to Scattered Data. *Computer Graphics and Image Processing*, (9):56–71, 1979.

[BS04] N. Bouaynaya and D. Schonfeld. A Complete System for Head Tracking Using Motion-Based Particle Filter and Randomly Perturbed Active Contour. *International Journal of Computer Vision*, page 18, 2004.

[Can09] James V. Candy. *Bayesian signal processing: classical, modern, and particle filtering methods*. John Wiley & Sons Ltd, Hoboken (N.J.), 1st edition, 2009.

[Cha] Sung-Hyuk Cha. Compreensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences*.

[DdFG01] Arnaud Doucet, J. F. G. de Freitas, and Neil Gordon. *Improving Regularised Particle Filters*. Springer-Verlag, 1st edition, 2001.

[DJ09] Arnaud Doucet and Adam M. Johansen. A Tutorial on Particle Filtering and Smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 2009.

[FPF99] Andrew Fitzgibbon, Maurizio Pilu, and Robert Fisher. Direct Least Square Fitting of Ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:476–480, 1999.

[GW08] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 3rd edition, 2008.

[IB98] Michael Isard and Andre Blake. CONDENSATION - Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[Kay93] Steven M. Kay. *Fundamentals of Statistical Signal Processing: estimation theory*. Prenticel-Hall, signal Processing series, 1993.

[KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[LC98] Jun S. Liu and R. Chen. Sequential Monte Carlo methods for dynamical systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.

[MK04] Gerard Medioni and Sing Bing Kang. *Emerging Topics in Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.

[Pap91] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes*. MacGraw-Hill, 3rd edition, 1991.

[Pet66] Anthony J. Pettofrezzo. *Matrices and Transformations*. Dover Publications, 1st edition, 1966.

[PS99] Michael K. Pitt and Neil Shephard. Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association*, 94(446):590–599, June 1999.

[PTVF07] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3rd edition, 2007.

[PYLJ09] Yi Peng, Qixiang Ye, Yanmei Liu, and Jianbin Jiao. Shaking Video Stabilization with Content Completion. *Proceedings of SPIE Electronic Imaging*, 7257(1), January 2009.

[SB91] Michael J. Swain and Dana H. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[Tek95] A. Murat Tekalp. *Digital Video Processing.* Prentice Hall signal processing series, 1995.

[Yan09] Junlan Yang. Robust Video Stabilization Based on Particle Filter Tracking of Projected Camera Motion. *IEEE Transactions on Computers*, EC 16(3), July 2009.