UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

CORNÉLIA JANAYNA PEREIRA PASSARINHO

UMA ABORDAGEM DINÂMICA PARA DETECÇÃO E SEGUIMENTO DE FACE EM VIDEOS COLORIDOS EM AMBIENTES NÃO CONTROLADOS

VITÓRIA
2012

CORNÉLIA JANAYNA PEREIRA PASSARINHO

UMA ABORDAGEM DINÂMICA PARA DETECÇÃO E SEGUIMENTO DE FACE EM VIDEOS COLORIDOS EM AMBIENTES NÃO CONTROLADOS

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Doutor em Engenharia Elétrica.

Orientador: Prof. Dr. Mário Sarcinelli Filho.

Orientador: Prof. Dr. Evandro Ottoni Teatini Salles

VITÓRIA
2012

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

Passarinho, Cornélia Janayna Pereira, 1977-
C566m Uma abordagem dinâmica para detecção e seguimento de face em vídeos coloridos em ambientes não controlados /
Cornélia Janayna Pereira Passarinho. - 2012.
109 f.: il.

Orientador: Mário Sarcinelli Filho.
Orientador: Evandro Ottoni Teatini Salles.
Tese (Doutorado em Engenharia Elétrica) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Processamento de imagens. 2. Visão por computador. 3. Kalman, filtragem de. 4. Vídeo digital. I. Sarcinelli Filho, Mário. II. Salles, Evandro Ottoni Teatini. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 621.3

CORNÉLIA JANAYNA PEREIRA PASSARINHO

UMA ABORDAGEM DINÂMICA PARA DETECÇÃO E SEGUIMENTO DE FACE EM VÍDEOS COLORIDOS EM AMBIENTES NÃO CONTROLADOS

Tese submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Doutor em Engenharia Elétrica - Automação.

Aprovada em 17 de dezembro de 2012.

COMISSÃO EXAMINADORA

Prof. Dr. Mário Sarcinelli Filho
Universidade Federal do Espírito Santo - Orientador

Prof. Dr. Evandro Ottoni Teatini Salles
Universidade Federal do Espírito Santo - Orientador

Profa. Dra. Raquel Frizera Vassallo
Universidade Federal do Espírito Santo

Prof. Dr. Klaus Fabian Côco
Universidade Federal do Espírito Santo

Prof. Dr. Thomas Walter Rauber
Universidade Federal de Espírito Santo

Prof. Dr. Adrião Duarte Dória Neto
Universidade Federal do Rio Grande do Norte

A Deus e a minha família.

Aos amigos.

Agradecimentos

Quero agradecer:

a Deus pela oportunidade de cursar o Doutorado na Universidade Federal do Espírito Santo, Programa de Pós-Graduação em Engenharia Elétrica, sob a orientação dos professores Evandro e Sarcinelli;

aos meus orientadores, pelo direcionamento na pesquisa e pela amizade;

às pessoas que permitiram que eu as filmasse e, assim, pudesse compor uma base de vídeos de teste na pesquisa desenvolvida neste doutorado;

aos professores Adrião, Thomas, Raquel e Klaus, por se disponibilizaram a avaliar esta Tese de Doutorado;

aos companheiros CISNEanos, aos amigos do LEPAC, LabSUL, LAIs e LabTel pelos cafés e agradáveis almoços no RU;

aos que me receberam em suas casas, como se fosse da família, em terras capixabas;

à professora Fátima Sombra, que me recebeu de volta no LabVIS, Universidade Federal do Ceará, em um intercâmbio extra oficial;

à minha família, que me apoiou incondicionalmente;

aos amigos e a todos que, de alguma forma, me apoiaram para que eu chegasse até aqui.

Valeu à pena!!

Soli Deo gloria.

Sumário

Lista de Tabelas

Lista de Figuras

Lista de Siglas

Seguem-se algumas siglas que são utilizadas ao longo deste texto. Algumas delas foram mantidas no idioma inglês, por serem mais conhecidas assim, o que facilitará a leitura do texto.

SDVS	Sistema Dinâmico com Vetores de Suporte
SVM	*Support Vector Machines*
HMM	*Hidden Markov Models*
RBF	*Radial Base Function*
ROC	*Receiver Operating Characteristic*
TFP	Taxa de falso positivo
TVP	Taxa de verdadeiro positivo
AUC	*Area Under the Curve*
FV	Face verdadeiro
FF	Face falso
NFV	Não-Face verdadeiro
NFF	Não-Face falso

Resumo

Detecção de face em imagens é uma linha de pesquisa que vem recebendo dedicação crescente em visão computacional, uma vez que pode ter diversas aplicações. Identificação, autenticação e reconhecimento de indivíduos são algumas tarefas realizadas a partir do desenvolvimento de sistemas que se baseiam em tais técnicas. O trabalho desenvolvido nesta Tese de Doutorado apresenta uma abordagem para detectar e seguir face em vídeos coloridos e adquiridos em ambientes não controlados. O Seguidor Dinâmico com Vetores de Suporte (SDVS) aqui proposto combina detecção de face com seguimento de alvo, através da integração de compensação de iluminação, detecção de cor de pele, classificador SVM (do inglês *Support Vector Machines*) usando características de Gabor e um filtro de Kalman discreto, compondo um sistema integrado. A arquitetura aqui proposta se distingue das demais encontradas na literatura por ser capaz de detectar e seguir face em vídeos adquiridos em ambientes externos, com condições do mundo real, ou seja, não controladas, detectar e seguir face em pose arbitrária, em diferentes tons de pele, e por apresentar robustez no que se refere à recuperação de falhas na detecção de face. Para validar a proposta do SDVS, foram realizados testes em vídeos das bases de dados de vídeos da Honda/UCSD e David Ross, como também vídeos capturados no Campus de Goiabeiras da Universidade Federal do Espírito Santo. Tais vídeos foram categorizados segundo o grau de dificuldade (desafio) a ser tratado, e os resultados da aplicação do SDVS foram comparados com uma técnica do estado da arte, para avaliar seu desempenho. Como resultado de tal comparação, pode-se concluir que a abordagem aqui proposta e o sistema SDVS implementado foram validados.

Abstract

Face detection in images is a growing branch of computer vision, regarding its potential for several applications. Identification, authentication and recognition of individuals are some tasks that are performed by systems that rely on such techniques. In this thesis, an approach is described to to detect and track a face in color uncontrolled videos. The Dynamic Support Vector Tracker (from the Brazilian Portuguese *Seguidor Dinâmico com Vetores de Suporte* - SDVS) *framework* here proposed combines face detection with target tracking, through integrating illumination compensation, skin color detection, Gabor features and a discrete Kalman filter, thus implementing an integrated system. Such architecture differs from others found in the literature for being able to detect and track faces in unconstrained outdoor videos, under real-world conditions, with different skin tones, tracking arbitrary poses of the face and for being capable of recovering failures in face detection. To validate the SDVS, tests were performed on videos from the Honda / UCSD and David Ross Video Databases, as well as videos captured at the Goiabeiras campus of the Federal University of Espírito Santo. These videos were categorized according to the degree of difficulty (challenge) to be treated, and the results of applying SDVS to them were compared with the correspondent results associated to a state of the art technique in order to evaluate the performance of the SDVS. The results suggest that the approach here proposed and the SDVS system implemented are validated.

Capítulo 1

Introdução

O seguimento de objetos é um problema fundamental na área de visão computacional (Yang et al.; 2011). A complexidade em seguir objetos pode surgir devido ao deslocamento abrupto do objeto, devido a mudanças no padrão de aparência do objeto e da cena, devido aos objetos serem de estrutura não rígida, devido a oclusões entre objetos ou oclusão do objeto pelo plano de fundo, devido a mudanças na iluminação da cena e devido a movimento da câmera. Estas situações tornam o seguimento um problema desafiador e ainda em aberto (Liu et al.; 2012).

O seguimento é realizado no contexto de aplicações que requerem a localização do objeto em cada quadro de uma sequência de vídeo. Geralmente, considerações são feitas para restringir o problema ao contexto de uma aplicação particular (Yilmaz et al.; 2006). Por exemplo, pode-se considerar que o objeto a ser seguido se movimenta suavemente e sem movimentos bruscos. Outras considerações típicas são quanto à velocidade constante e à aceleração a partir de informações a priori. Informações prévias a respeito do número, dimensão, aparência e forma do objeto também podem ser levadas em conta para simplificar o problema de seguimento.

Os métodos de seguimento diferem entre si pela forma como abordam as seguintes questões (Yilmaz et al.; 2006): que representação do objeto é adequada para o seguimento? Que características da imagem deveriam ser utilizadas? Como o movimento, a aparência e a forma do objeto devem ser modelados? As respostas a estas questões dependem do contexto ou ambiente em que o seguimento é feito, e do objetivo para o qual as informações estão sendo procuradas na sequência de vídeo.

Além do interesse intrínseco ao problema, seguimento de objetos tem sido alvo constante de pesquisas científicas pela vasta possibilidade de aplicação no seguimento de face humana. Métodos de identificação e de monitoramento de pessoas sempre foram

muito importantes para toda a sociedade. No mundo moderno, as pessoas normalmente precisam carregar documentos para quaisquer lugares que forem, pois essa é a única forma de provarem suas identidades (Zhao et al.; 2003). Por outro lado, o monitoramento remoto de pessoas, utilizando imagens tomadas por câmeras de segurança, tem adquirido uma importância significativa, principalmente em casos de áreas com grande número de pessoas.

Assumindo-se que não existem pessoas completamente idênticas, a necessidade da utilização de tais documentos extingue-se quando se dispõe de métodos capazes de diferenciar cada indivíduo sem confundi-lo com seus semelhantes. Seguramente, esse é o principal objetivo da pesquisa em biometria, cujo foco principal tem sido a verificação e identificação de pessoas utilizando propriedades biológicas dessas pessoas (Yanushkevich; 2006). O seguimento de uma pessoa em atitude suspeita ao longo de um vídeo capturado por uma ou mais câmeras de segurança também é uma atividade que tem ganhado muito interesse, principalmente no que tange à segurança pública.

Dentre as técnicas de reconhecimento biométrico de pessoas que são utilizadas atualmente, as mais precisas são aquelas baseadas em imagens do fundo da retina e as baseadas em imagens de íris (de Campos; 2001; Pankanti et al.; 2000; Ratha et al.; 2001). A confiabilidade de sistemas de reconhecimento de íris é tão grande que algumas instituições financeiras os adotam para identificar seus usuários. Porém, essas abordagens têm o problema de serem um tanto quanto invasivas, pois para o funcionamento dos sistemas atuais é necessário impor certas condições ao usuário. No caso dos sistemas de reconhecimento por imagem de íris, por exemplo, o usuário deve permanecer parado, em uma posição definida e com os olhos abertos, enquanto uma fonte de luz ilumina os olhos e um scanner de íris ou uma câmera captura a imagem. O caráter invasivo acentua-se ainda mais em sistemas que utilizam imagens de fundo de retina, uma vez que, atualmente, é preciso utilizar um colírio para dilatar a pupila do usuário antes de efetuar a aquisição da imagem. Nesse ponto está a mais sobressalente vantagem de um sistema de reconhecimento baseado em imagens de faces: ele é não invasivo.

A pesquisa em reconhecimento de faces vem se desenvolvendo no sentido da criação de sistemas capazes de detectar (e identificar, se for o caso) pessoas mesmo quando essas não percebam que estão sendo observadas. Dessa forma, é possível que, no futuro, uma criança desaparecida seja localizada através de imagens de câmeras localizadas em pontos estratégicos de uma cidade, como estações de metrô e cruzamentos de avenidas.

Várias outras aplicações motivadoras para a pesquisa nessa área podem ser citadas, como (de Campos; 2001):

- identificação pessoal para acesso restrito a dados bancários, dados referentes a passaporte, fichas criminais, etc.;
- sistemas de segurança e controle de acesso;
- monitoramento de multidões em estações e shopping centers;
- criação de retrato falado;
- busca em fichas criminais;
- envelhecimento computadorizado para auxiliar a busca por desaparecidos, e
- interfaces perceptuais homem-máquina com reconhecimento de expressões faciais.

O reconhecimento de faces é uma área de pesquisa desafiadora, que abre portas para a implementação de aplicações muito promissoras. Embora muitos algoritmos eficientes e robustos já tenham sido propostos, ainda restam vários desafios a serem abordados e solucionados.

Nesta Tese de Doutorado, em particular, se propõe um sistema de detecção e seguimento de face em sequências de vídeo capturadas em ambientes internos e externos, não controlados, através de uma representação compacta, invariante às condições de iluminação, ao plano de visão, ao movimento da imagem e a alterações de expressões, que possibilite distinguir indivíduos em tempo hábil.

1.1 Caracterização do Problema

Segundo Li e Jain (2005, p. 395) os resultados do reconhecimento de face são altamente dependentes das características que são extraídas para representar o padrão de face e dos métodos de classificação utilizados para distinguir entre faces. Por sua vez, localização e normalização são a base para extração de características efetivas. Um sistema de reconhecimento de face geralmente consiste de quatro módulos, como pode ser visto no diagrama apresentado na Figura 1.1.

A detecção de face segmenta as áreas de face do plano de fundo da imagem. No caso de vídeo, a face detectada pode precisar ser seguida, utilizando-se um componente seguidor de face. Em caso de oclusão, o sistema de detecção é novamente ativado, até que se tenha um alvo a seguir. O sistema só é iniciado quando a primeira face é detectada. Além disto, o ideal é que haja uma estratétgia de recuperação de falha no

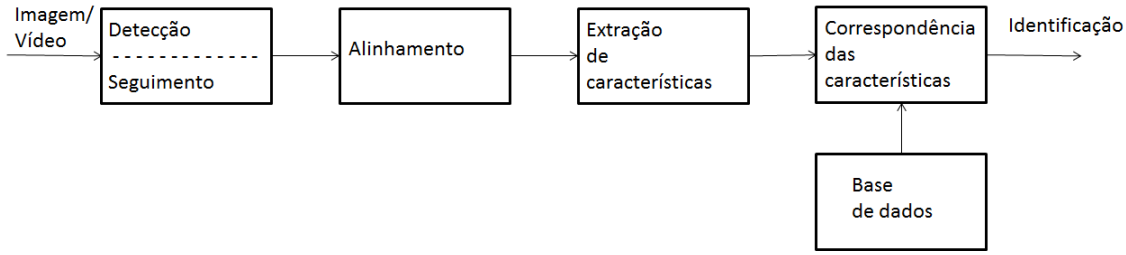


Figura 1.1: Fluxo de processamento para detecção e reconhecimento de face.

seguimento de face, em situações de perda da localização do alvo. Alinhamento de face visa conseguir localização e normalização mais precisa das faces, considerando que o passo de detecção de face fornece estimativas grosseiras da localização e dimensão de cada face detectada. Componentes faciais, como olhos, nariz, boca e contorno facial, estão localizados. Com base nestes pontos de localização, a imagem da face de entrada é normalizada em relação a tamanho e postura, utilizando transformações geométricas. A face é também normalizada com respeito às propriedades fotométricas, tais como iluminação e nível de cinza. Finalmente, após a normalização, a extração de características é realizada, para prover informação efetiva e invariante, geométrica e fotometricamente, para distinguir entre faces de diferentes pessoas.

O desempenho no seguimento e detecção de face em cenas dinâmicas é dependente da precisão do estágio de detecção do alvo. Os principais métodos de detecção de face são divididos em métodos que utilizam cor da face, métodos que detectam características de faces e métodos holísticos. Estes métodos são utilizados tanto para reconhecimento como para identificação de indivíduos em vídeo.

A detecção de faces combinando métodos baseados em cores pode garantir alto desempenho. As vantagens são que tais métodos são rápidos e têm alta taxa de detecção. Um dos métodos mais conhecidos é o seguidor de faces elíptico (Birchfield; 1997; Nummiaro et al.; 2002). Neste método, é necessário indicar, no vídeo, o alvo a ser detectado. A partir de então, a localização do objeto que segue a distribuição de cor do alvo é estimada por um filtro Bayesiano. Mais recetemente a cor tem sido utilizada juntamente com outras características para rastrear face, como os métodos apresentados em (Lee and Kim; 2007; Lin; 2007; Do et al.; 2007) que utilizam especificamente a cor da pele assosciada à detecção de características da face e sua distribuição geométrica.

Contudo, abordagens que utilizam somente a distribuição de cor têm alguns inconvenientes, pois tais métodos são limitados na presença de iluminação variante. Por isto, carecem de uma estratégia que garante algum grau de constância de cor. Além disso, os detectoes desenvolvidos para esse fim podem encontrar objetos que tenham uma cor similar à cor do alvo, e assim gerar um falso positivo.

Nos métodos baseados em características para detectar face é feita uma busca por uma ou mais características, como olhos, nariz e boca. Configurações plausíveis entre as características detectadas poderiam, então, ser identificadas como faces. Castañeda, Luzanov e Cockburn (2004) utilizam uma combinação de características para detectar a presença de face no vídeo. A candidata a face é localizada na região de pele identificada na imagem. Nesta região são verificadas a presença das características da face utilizando um SVM (do inglês *Support Vector Machine*) para detectar a boca e um SVM para detectar os olhos. São permitidos aos indivíduos apenas movimentos suaves em frente à câmera, numa faixa de mais ou menos 10 graus. Já Campos, Feris e Cesar Junior (2000) propõem uma arquitetura para identificação em vídeo que utiliza os dois olhos, a boca e o nariz como características a serem detectadas e seguidas.

Especificamente, métodos de detecção baseados em características demandam alto esforço computacional e operações em baixa velocidade. Além disso, o problema principal destes métodos é que requerem um detector de olho, um detector de nariz, um detector de boca, e assim por diante. Neste casos, o problema de detectar faces é substituído pelo problema de múltiplas detecções, similarmente complexo, de partes deformáveis. Contudo, o conjunto de detectores pode reduzir o desempenho obtido com cada detector de característica facial, e o sistema deveria realizar múltiplas tarefas de detecção (Gong et al.; 2000).

Tais métodos são úteis para análise de face e correspondência na identificação facial, pois a detecção e alinhamento das características faciais demandam imagens de resolução espacial relativamente alta, no mínimo 128×128. Além disso, em cenas dinâmicas em ambientes com restrições reais, com plano de fundo complexo, a detecção de faces necessitará ser realizada, muitas vezes, em imagens de resolução muito baixa. Oclusões causadas por mudanças no campo de visão são o principal problema com os métodos baseados em características. Em tais casos, a correspondência entre certas características não existe, devido à oclusão. É o caso de abordagens que usam características dos olhos, tais como o globo ocular, a parte branca do olho, ou até mesmo a pupila (Gong et al.; 2000), pois gera-se falsa detecção quando a pessoa fecha os olhos ou usa óculos escuros.

Nos métodos holísticos nenhuma subparte semanticamente significativa da face é definida, e a face não é decomposta. O objetivo é evitar a decomposição arbitrária da face em tais partes. A detecção é tratada como uma busca pela face em sua inteireza. O arranjo espacial das características da face é implicitamente codificado dentro da representação holística das estruturas internas da face. Neste tipo de representação é possível tratar imagens de resoluções relativamente baixas (Li and Jain; 2005; Gong et al.; 2000; Belhumeur et al.; 1997). Turk e Pentland (1991) utilizam *eigenfaces* para

reconhecimento de faces, o que é um exemplo de utilização da representação holística da face.

O seguimento de face é o meio utilizado para identificar ou reconhecer indivíduos em sequências de quadros de um fluxo de vídeo. O passo de seguimento de face, na maioria dos trabalhos estudados, é feito utilizando-se estimação estatística. É uma escolha natural quando o problema em questão apresenta a necessidade de estimação de localização num sistema que muda com o tempo, com observações ruidosas. Por exemplo, Chellappa e Zhou (2002) propõem um método Bayesiano para reconhecimento de pessoas em vídeo. Já Zhao, Chellappa e Phillips (2002) propõem um modelo probabilístico parametrizado pelo vetor de estados de seguimento, simultaneamente caracterizado pela cinemática e identidade de pessoas. Outros autores que utilizam o seguimento adaptativo em vídeos reais são Kim, Kumar e Pavlovic (2008), que utilizam um modelo HMM (do inglês *Hidden Markov Models*) e o método LDA (do inglês *Linear Discriminant Analysis*) para reconhecimento de indivíduos. Já Seo (2009) apresenta um estudo sobre seguimento e reconhecimento de faces utilizando um filtro de partículas.

Uma categoria de algoritmos para seguir faces que é relevante ser ressaltada está associada ao método CAMSHIFT (Comaniciu and Meer; 1997). Em (Yao and Gao; 2000) é proposto um seguidor baseado na pele e na transformada de cor. Em (Huang and Chen; 2000) é apresentado um seguidor de faces para um modelo estatístico de cor e um modelo deformável para múltiplas faces. O MeanShift também é mencionado num trabalho em que é proposto seguir o gradiente ascendente na distribuição de cor da pele (Dadgostar et al.; 2005). Já em (Terrillon et al.; 2004) a cabeça é modelada como um mapa de textura 3D da imagem. Note-se que tais métodos têm se mostrado rápidos e adequados a problemas que demandam tempo real.

É preciso salientar que, em sua maioria, estes métodos utilizam um modelo estatístico de cor para seguir a face. Como discutido anteriormente, porém, na presença de objetos de fundo com cor similar ao alvo é esperado que eles falhem.

A literatura traz um método proposto por Viola e Jones (2001), considerado *top* no estado da arte, para detecção de faces. Até então, a melhor solução para detecção de faces humanas era através de um *template* (um molde de face) e uma medida de diferença para identificar o padrão face do padrão não face em imagens em nível de cinza (Sung and Poggio; 1998). O método de Viola e Jones é um detector de face proposto para ser rápido, e por esta razão tem sido usado em sequências de vídeo, analisando cada quadro do vídeo de forma independente dos anteriores, ou seja, considerando cada quadro do vídeo como se fosse uma imagem isolada, sem, portanto, lidar com a dinâmica

da cena. Ele detecta faces frontais em imagens em nível de cinza, através da busca exaustiva de características de Wavelets de Haar (Mallat; 2008). Estas características são usadas para treinar um conjunto de classificadores operando em cascata, sendo que a região analisada é considerada como sendo face se esta hipótese não for rejeitada por nenhum desses classificadores.

O seguimento de face é um assunto importante em muitos problemas de visão computacional, particularmente na área de vigilância em vídeo (Pentland and Choudhury; 2000; de Almeida; 2006). Seguimento de face estende a detecção de face em uma imagem estática para uma sequência de vídeo, em que informação espaço-temporal pode ser utilizada. Para desempenhar a tarefa de seguir face espera-se que sistemas de seguimento sejam robustos a falhas (Toyama and Hager; 1997). Isto significa dizer que o algoritmo de seguimento de face deve ser capaz de desempenhar o seguimento apesar dos inconvenientes comuns a sistemas de visão computacional, especialmente em ambientes reais.

Mesmo que o seguimento de face seja robusto a falhas, podem ocorrer eventos que causem a falha do sistema, como, por exemplo, oclusão prolongada da face. Devido a esta possibilidade de falha, espera-se que um sistema de seguimento de face apresente robustez, ou capacidade de recuperação em caso de falhas (Toyama; 1998). No pior caso, a falha no seguimento implica em ter nenhuma informação sobre a face alvo. A recuperação, então, se torna equivalente à inicialização do seguimento, em que a detecção da face, e, em seguida, o reconhecimento da face, são necessários para localizar e verificar o alvo antes que o seguimento possa continuar.

1.2 Objetivos desta Tese

O objetivo geral desta Tese de Doutorado é propor um sistema para seguir uma face em pose arbitrária, em diferentes planos, detectada a partir de um modelo holístico utilizando um banco de filtros, considerando vídeos adquiridos em ambientes não controlados. Neste método de seguimento, em que os vídeos coloridos são tomados em ambiente que apresente ou não algum grau de controle da iluminação, um filtro recursivo discreto, associado a um classificador binário, é robusto para recuperar a localização de uma face não detectada. O sistema proposto utiliza o resultado do classificador binário, a detectão de face, como observação para o filtro recursivo. Em situações em que não há obervação para o filtro, a face não é detectada, a abordagem aqui proposta é robusta a continuar o seguimente para o próximo quadro. O seguidor proposto nesta Tese é denominado Seguidor Dinâmico com Vetores de Suporte (SDVS).

Além disto, o SDVS detecta e segue a face levando em consideração o tempo requerido à dinâmica do problema.

Assim, os objetivos específicos desta Tese são:

- abordar o problema de captura de vídeos em ambientes não controlados por meio da adaptação de um algoritmo de constância de cor, que seja pouco sensível a nuances de tons de pele, e que não necessite aplicar uma transformação de espaço de cor, tornando o método mais direto;
- utilizar vetor de características de Gabor para treinar o classificador para detectar a face, representação matemática que permite caracterizar os padrões face em direfentes graus de orientação, escala e translação, além de ser suficiente para futuro passo de reconhecimento da face detectada;
- utilizar uma técnica matemática para extrair características para detetar face que seja suficiente para identificar, em trabalhos futuros, os indivíduos nas sequências de vídeo.

1.3 Contribuição desta Tese

A contribuição desta Tese, considerando os objetivos propostos, é uma abordagem que:

- (a) considera a constância de cor, problema que pode afetar o desempenho de seguimento em vídeos coloridos gravados em ambientes não controlados, utilizando um algoritmo que usa diretamente os valores de pixel no espaço de cor RGB; Apesar de alguns trabalhos apresentarem solução de seguimento de vídeos coloridos robusto à iluminação, não utilizam um preprocessmento que trate efetivamente o problema da constância de cor. Na verdade, eles fazem uso de imagens em nível de cinza juntamente com características de face ditas invariante à iluminação. O agoritmo de constância de cor aplicado diretamente ao espaço RGB não normalizado livra o preprocessamento do passo de mudança de espaço de cor.
- (b) propõe a identificação de uma faixa de tons de pele em que estão incluídos os tons de pele escuros, não identificados em ambientes não controlados. A detecção de uma faixa de cor de tons de pele é realizada no espaço de cor RGB, etapa que só é possível após a utilização de uma algoritmo de constância de cor. Em

sistemas que são utilizados para seguir face em sequências de vídeos coloridos em um ambiente não controlado a identificação de regiões de pele é um importante atributo, pois restringe as regiões de busca de candidatas a face. A identificação de pele também pode ser utilizada para identificar que a face encontrada é de um humano. A literatura apresenta uma faixa de valores rígida para os tons de pele humana, porém, os resultados alcançados nesta tese mostraram que esta faixa de valores funciona somente em imagens com algum controle de iluminação. Mesmo com a transformação da imagem para o espaço de cor HSV (do inglês *Hue*, *Saturation* e *Brightness*), que a literatura apresenta como invariante à iluminação, a faixa de valores do estado da arte apresenta resultado inferior ao método proposto.

- (c) detecta e segue de maneira robusta a face numa sequência de vídeo, graças à sua capacidade de recuperação de uma face não detectada em um determinado quadro da sequência, em poses e orientações arbitrárias. Em casos de seguimento de falsos positivos, é possível, ao longo do fluxo de vídeo, identificar que o alvo seguido não é uma face. Isto é possível pois a detecção de face pelo SVM é utilizada a cada instante de tempo para confirmar que o objeto localizado na posição estimada pelo Filtro de Kalman é um verdadeiro positivo ou não. O que a literatura apresenta são soluções de seguimento que esperam o surgimento da face, votlando ao passo inicial. A face é detectada e a tragetória da localização deste alvo é seguido sem que haja a verificação de que é um verdadeiro positivo ao longo do rastreio.

1.4 Estrutura do Texto

Este Texto está estruturado na forma de capítulos, assim dispostos:

Introdução - este capítulo, em que se descreve o tema e o problema a ser abordado. É apresentada a motivação e a caracterização do problema, o objetivo e a contribuição original deste trabalho. Também é apresentada uma revisão bibliográfica a respeito dos temas estudados ao longo da pesquisa para elaboração da Tese, a saber: técnicas de detecção e seguimento de faces, o problema da má iluminação em ambientes não controlados e técnicas de compensação de iluminação, e seguimento de face em seqüências de vídeos coloridos.

Metodologia Adotada - nesse capítulo é apresentada a metodologia proposta para seguir face em vídeos coloridos gravados em ambientes não controlados. A partir dos

estudos feitos são apresentados os métodos escolhidos para resolver o problema proposto e para implementação do *framework* SDVS.

Seguidor Dinâmico com Vetores Suporte - a maneira como as técnicas apresentadas na Metodologia Adotada cooperam entre si na resolução do problema em questão é apresentada neste capítulo. O código gerado para implementar o *framework* SDVS está disponível no Apêndice B.

Resultados - os resultados dos testes feitos com o sistema de seguimento de face proposto são abordados neste capítulo, e são exaustivamente analisados. Foram feitos testes com vídeos capturados no laboratório CISNE e em ambientes externos, sempre no Campus de Goiabeiras da UFES, e também com seqüências de vídeo obtidas a partir de bancos de dados reconhecidos pela comunidade científica. O capítulo conclui ressaltando que os resultados obtidos validam a metodologia adotada e o sistema de seguimento de face implementado.

Conclusões e Trabalhos Futuros - neste capítulo são apresentadas as considerações a respeito dos resultados alcançados ao longo da pesquisa para elaboração desta Tese, inclusive vislumbrando trabalhos futuros.

Capítulo 2

Base Teórico-matemática para a Metodologia Adotada

No Capítulo 1 foi apresentada uma revisão bibliográfica a respeito do tema abordado nesta Tese de Doutorado. De acordo com o estudo realizado, foram elencados os métodos utilizados com o intuito de resolver o problema de detecção e seguimento de face em sequências de vídeo. Como mencionado em tal capítulo, neste capítulo são descritas as técnicas de processamento de imagens e reconhecimento de padrões para a metodologia proposta para detectar e seguir face em vídeos adquiridos em ambientes não controlados.

O desempenho no seguimento e detecção de face em cenas dinâmicas é dependente do desempenho do estágio de detecção do alvo e dos estágios anteriores a este. Em primeiro lugar, um método para contornar a presença de má iluminação deve ser aplicado, com o fim de tornar o sistema invariante às características fotométricas do ambiente em que as sequências de vídeo são capturadas. Além disso, a abordagem utilizada para a detecção da face deve apresentar baixas taxas de falsos positivos. Note-se que isto é extremamente difícil, pois existem regiões da imagem que, embora não sendo regiões de face, quando observadas fora do contexto global são admitidas como regiões de face. Entretanto, é importante frisar que o número de falsos positivos pode ser muito reduzido realizando a busca da face somente nas regiões da imagem em que é provável encontrar pixels de face, estratégia esta que é adotada nesta Tese.

2.1 Iluminação e Constância de Cor

De acordo com a estratégia adotada nesta tese, admite-se que seja mais provável identificar um pedaço de uma imagem como sendo uma face humana em regiões da imagem que apresentem pixels cujos valores estejam dentro da faixa de cores de tons de pele humana. Assim, é necessário que um sistema para seguimento de face seja capaz de detectar tons de pele, mesmo sob condições naturais de mudança de iluminação. Para tanto, é adequado a utilização de um algoritmo capaz de amenizar aqueles efeitos, produzindo uma imagem com cores próximas às da cena real, capturada em vídeo.

Com o intuito de se propor um seguidor de face eficiente, foi utilizado um método para compensação de iluminação, que é o primeiro estágio no tratamento de cada quadro da sequência de vídeo sob análise. Tendo em conta o fato que as sequências de vídeo aqui analisadas foram gravadas em ambientes não controlados, se faz necessária uma etapa de normalização das imagens, já que a iluminação do ambiente não era (é o caso de vídeos adquiridos em ambientes externos, por exemplo). Observe que, como a iluminação influencia na cor do objeto de interesse, a etapa de compensação de iluminação é necessária para normalizar qualquer sequência de imagens em que a metodologia proposta seja utilizada, sendo o método de constância de cor das imagens adotado nesta Tese apresentado na sequência desta seção.

A habilidade do sistema de visão humano de corrigir naturalmente a cor dos objetos, apesar da cor da fonte de luz, é conhecida como constância de cor. O mesmo processo não é trivial para sistemas de visão de máquina em cenas reais. Por esta razão, constância de cor é uma das áreas de pesquisa mais importantes nas áreas de visão computacional e processamento de imagens coloridas, com uma vasta gama de aplicações. Entretanto, ainda não foi identificada uma solução única para o problema. O alvo da pesquisa em constância de cor é produzir uma descrição da cena invariante ao iluminante, quando ela é tomada sob iluminação desconhecida/variante.

Matematicamente, uma imagem colorida é representada como

$$E_k(x,y) = \int_{\omega} R(x,y,\lambda)\, L(\lambda)\, S_k(\lambda)\, d\lambda, \tag{2.1}$$

em que $R(x,y,\lambda)$ é a reflectância da superfície, $L(\lambda)$ é a propriedade da iluminação e $S_k(\lambda)$ é a característica do sensor (como função do comprimento de onda λ), sobre o espectro visível ω. O índice k representa a resposta do sensor no k-ésimo canal e $E_k(x,y)$ é a imagem correspondente ao k-ésimo canal ($k \in \{R,G,B\}$).

Também, é bem estabelecido que um problema é mal posto quando uma ou mais das seguintes três condições não se verificam (Agarwal et al.; 2006): (a) existe uma

solução, (b) esta solução é única, e (c) esta solução única é estável. Em constância de cor, a unicidade e a estabilidade da solução não podem ser garantidas, devido à alta correlação entre a cor da imagem e a cor do iluminante, ou seja, trata-se sempre de um problema mal posto.

A maioria dos algoritmos de seguimento de face requer que este seja desempenhado em tempo real, sob condições de iluminação irrestritas, e em um ambiente dinâmico. Estes requerimentos demandam que tais algoritmos se adequem à variação de cor do alvo, mudança do plano de fundo, oclusões e movimento arbitrário do alvo a ser seguido. Neste contexto, a constância de cor pode ser aplicada para contornar a influência da mudança de iluminação em aplicações de seguimento de vídeos coloridos, o que é feito nesta Tese de Doutorado. Porém, as abordagens de implementação podem ou não ser adequadamente aplicadas ao problema de seguimento de vídeo, quer por demandarem grande esforço computacional, quer por demandarem a escolha de parâmetros ótimos, como é o caso dos métodos baseados no algoritmo Retinex (Ebner; 2007).

Para garantir a adaptabilidade e robustez do algoritmo de seguimento de face às condições de um ambiente real foi utilizado o algoritmo de constância de cor baseado no método proposto em (Chen and Grecos; 2005), que é uma modificação do método *Grey World* (Ebner; 2007). O método não depende de limiarização nem necessita de mudança de espaço de cores, utiliza os valores dos pixels diretamente no espaço RGB (do inglês *Red Green Blue*) da imagem a ser processada. Além disto, o método se mostra rápido o suficiente para ser utilizado em um sistema de seguimento em tempo real. Diferentemente do método original, porém, a versão empregada nesta Tese não utiliza transformações para outros espaços de cores, ou seja, as imagens são tratadas no próprio espaço RGB (do inglês *Red Green Blue*). Originalmente é utilizado o espaço de cor RGB normalizado.

A cor das cenas capturadas em vídeo pode ser modelada pela relação intravável em (2.1). Diante disto, assume-se, nesta tese, a utilizado um algoritmo para prover constância de cor sob a condição de que a minimização dos efeitos da iluminação é feita a cada quadro da sequência de vídeo. A escolha do espaço RGB se justificada através dos resultados alcançados pela da observação dos resultados mostrados na Figura 2.1, além dos resultados com a detecção de pele. A identificação cor real da cena é uma questão subjetiva, pois está sujeita a observação por um indivíduo, o que dificulta a comparação com resultados de algoritmos de constância de cor.

O método utilizado se baseia no fator de escala

$$S = \frac{C_{std}}{C_{avg}}, \tag{2.2}$$

para um canal de cor específico (R, G, B), e C_{std} é o valor de cinza médio padrão, obtido pela equação

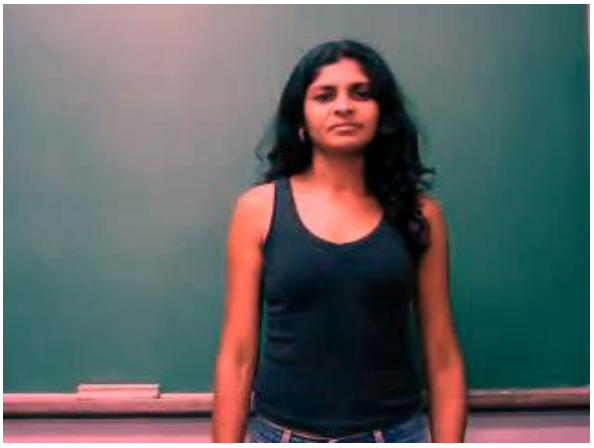
$$C_{std} = \frac{\sum_{i=1}^{m} [max(N_R, N_G, N_B) + min(N_R, N_G, N_B)]}{2n}, \qquad n = m - \sum_{i=1}^{m} l, \tag{2.3}$$

onde $l = 1$ se $N_R = N_G = N_B = 0$ (ou seja, se o pixel é preto) e $l = 0$ no caso contrário, m é o número de pixels na imagem, e n é o número de pixels não pretos (aqui se deseja contornar o problema de compensação de imagens que tenham grande parte dos seus pixels pretos). Por sua vez, C_{avg} é o valor médio do canal específico.

Na Figura 2.1 é mostrada uma imagem correspondente a um dos quadros de um dos vídeos usados como teste do sistema de seguimento de face proposto nesta Tese. Na parte superior da figura é mostrada a imagem crua (sem compensação de ilumina-ção), enquanto na parte inferior é apresentada a mesma imagem, após a aplicação do algoritmo de compensação de iluminação.

No método proposto em (Chen and Grecos; 2005) o valor C_{std} depende somente dos valores dos pixels da imagem original, não sendo necessário escolher parâmetros ótimos. No algoritmo original *Grey World*, porém, o valor C_{std} é constante $(0,5)$, o que não é adequado para imagens sujeitas a mudanças de iluminação. Imagens exemplificando o uso da versão aqui implementada do algoritmo em (Chen and Grecos; 2005) são exibidas na segunda coluna da Figura 2.1, enquanto que os mesmos exemplos, utilizando agora o algoritmo Retinex, são mostrados na terceira coluna da figura. Destaque-se, mais uma vez, que o método de constância de cor escolhido não utiliza algoritmos de ordenação nem mudanças de espaço de cor, e sua escolha decorre do compromisso entre eficiência e baixo custo computacional. Observe que quando da aplicação do algoritmo Retinex (Figura 2.1, terceira coluna) em imagens com plano de fundo escuro e objetos de interesse claros, estes objetos se tornam mais claros do que o necessário. Este fenômeno também poderia ocorrer no algoritmo *Grey World* original, pois ali o valor C_{std} é constante (igual a $0,5$). Porém, como se pode notar a partir da terceira coluna da Figura 2.1, tal fenômeno não ocorre quando a versão do algoritmo de compensação de iluminação em (Chen and Grecos; 2005) aqui proposta é utilizada.

(a) Imagem original.



(b) A imagem após a compensação de iluminação.

Figura 2.1: Exemplificando o efeito do algoritmo de compensação de iluminação.

2.2 Cor da Pele

Após a etapa de compensação de iluminação, é utilizado um algoritmo de detecção da face alvo, baseado nas regiões de cor da pele detectadas na imagem sob análise. Assim, um filtro de cor de pele é considerado neste trabalho. Tal método é importante para remoção de pixels que não são candidatos a face, reduzindo assim a região de busca para a etapa de detecção de face.

A cor da pele é um atributo utilizado no seguimento de face. Assim, é importante obter uma constância de cor aproximada da cor da face. Se isto é conseguido em tempo real, a complexidade das etapas seguintes do seguimento é diminuída. A cor da pele é invariante à movimentos de rotação e translação e escala. Os pixels de tom de pele podem diminuir a região de localização da posição da face no seguimento, além de garantir que a face detectada naquela região pertença a um indivíduo humano.

Figura 2.2: Constância de cor: imagens originais (primeira coluna), resultado obtido com o algoritmo de constância de cor utilizado (segunda coluna) e resultado obtido usando o algoritmo Retinex (terceira coluna).

O problema primário na detecção automática da pele é a constância de cor. A cor do pixel de uma imagem depende não apenas da cor do objeto de interesse, como mostra a Equação (2.1) (Seção 2.1). Isto significa que o valor da cor do pixel de um pedaço de pele pode ser muito diferente, dependendo da câmera utilizada para capturar a imagem, da cor e da intensidade da iluminação, da orientação relativa entre a câmera, a superfície da pele e a fonte de luz, ou da existência de sombras ou realces na imagem. Devido à complexidade da iluminação quando se considera vídeos adquiridos em ambientes do mundo real, um algoritmo de constância de cor é indispensável para a detecção robusta de tons de cor da pele. Diante destas restrições, é utilizado nesta Tese de Doutorado o método de compensação de iluminação descrito na Seção 2.1, para detecção robusta de regiões da imagem com cor de pele.

Figura 2.3: Detecção de pele: imagens originais (primeira coluna) e respectivas regiões da imagem detectadas como pele (segunda coluna), região de pele é identificaca com pixels branco.

Para detecção de pele, o algoritmo apresentado em (Peer et al.; 2003), proposto para detectar a região de pele em imagens coloridas, usa limiares nos valores RGB de cada pixel na imagem para identificar as regiões de pixels de cor de pele. Os limiares aplicados a cada pixel são 95 para R, 40 para G e 20 para B, obtidos a partir de testes feitos com imagens de indivíduos com cor de pele clara a parda como os indivíduos da Figura 2.2. Então, se a diferença absoluta entre os valores de R e G é maior que 15 e os valores de R são maiores que os vales nos canais G e B, o pixel é classificado como pele. Contudo, é relevante ressaltar que o algoritmo de identificação de pixels que tenham cor de pele não funciona bem sem o estágio anterior de constância de cor, conforme nossos experimentos preliminares mostraram. Não obstante, os limiares propostos na literatura não se mostraram eficientes para detecção de tons de pele muito escuros como o do indivíduo da Figura 2.2, mesmo utilizando o algoritmo de compensação de iluminação. Assim, foi necessária a mudança daqueles limiares para 20 para R, 30 para G e 100 para B e eliminação da condição de diferença absoluta entre os valores de R e G é maior que 15. Chegou-se a estes valores através de testes de diversos limiares utilizando as imagens capturadas dos vídeos produzidos durante esta pesquisa. São

Figura 2.4: Imagens com indivíduos que tem diferentes tons de pele.

Figura 2.5: Imagens com indivíduo com tom de pele escuro.

vídeos com pouco ou nenhum controle de iluminação e os indivíduos tem desde tom de pele clara até tom de pele muito escuro, como o das Figuras 2.2 e Figura 2.2. A condição de os valores de R serem maiores que os valores nos canais G e B permane. Os valores de R são mais mais altos do que os valores da outras bandas nos diferentes tons de pele.

A Figura 2.2 exibe exemplos de detecção de pele em imagens de um banco de imagens de faces de pele escura, quadros avulsos de vídeos de teste gravados em vários locais no campus da UFES, e imagens de celebridades conseguidas na internet. Note-se que em alguns casos regiões das imagens que são detectadas como pele não o são efetivamente (ver o boné na cabeça do indivíduo no quadro da segunda linha da figura), mas são objetos com cor semelhante aos tons de pele humana. Há também regiões que são de pele, mas não foram detectadas pelo algoritmo de detecção de pele. As falsas regiões de pele, bem como as regiões de pele que não são detectadas, acontecem mesmo com a utilização do algoritmo de constância de cor, confirmando que é difícil a tarefa de identificação de pele em imagens coloridas adquiridas em ambientes reais.

É importante salientar que foram feitos testes com outros modelos de cor, como YCbCr (Y, Cb, Cr representam, respectivamente, luminância, componente da crominância da diferença de azul e componente da crominância da direfença de vermelho) e HSV, na tentativa de escolher um método para compensação de iluminação e detecção de pixels de cor da pele. No entanto, os testes feitos no espaço RGB apresentaram, sob avaliação visual, os melhores resultados de detecção de pele para as imagens capturadas

em ambientes não controlados, tanto internos quanto externos. Dessa forma, não são feitas quaisquer transformações de espaço de cores nesta Tese de Doutorado.

2.3 Filtro de Gabor

A face a ser detectada e rastreada nos vídeos experimentais desta pesquisa pode não aparecer em posição vertical ou mesmo em escala constante. A face pode não estar completamente visível devido a oclusões. Os idivíduos podem se mover livremente, como é previsível em um ambiente de circulação de pessoas. Um dos objetivos a serem alcançados nesta pesquisa de doutorado prover é uma representação de imagem que permitisse a caracterização da face em diversas poses, escalas e oclusão parcial.

Diante deste desafio, foram escolhidas as respostas de um filtro de Gabor. Como demonstrado em (Lee; 1996), representação originalmente extendida de 1D para 2D por Daugman (Daugman; 1985), as respostas de Gabor são eficientes para detecção e reconhecimento de objetos 2D. Isto se deve ao fato de que, utilizando o filtro de Gabor, é possível representar um objeto de interessse de uma imagem, ainda que esta esteja rotacionada ou em diferentes escalas. Calibrando-se desvio padrão, frenquência e orientação é possível obter a representação de uma face em diferentes poses e escalas, como descrito a seguir.

O filtro de Gabor é definido como uma onda senoidal modulada por uma Gaussiana, conforme a equação

$$\psi(\mathbf{p}) = \frac{1}{2\sigma_x\sigma_y} exp\left[\left(\frac{-\|\mathbf{p}\|^2}{2\sigma_x\sigma_y}\right)\right] exp\left[j2\pi\left(\mathbf{w}^T(x,y)+\phi\right)\right], \tag{2.4}$$

e sua transformada de Fourier

$$\Psi(u,v) = \exp\left[-2\pi\left((u-u_0)^2\sigma_u^2 + (v-v_0)^2\sigma_v^2\right)\right], \tag{2.5}$$

em que $j=\sqrt{-1}$, $\mathbf{p}=(x_r,y_r)$, $\mathbf{w}^T=(u_0,v_0)$ é a frequência no plano de onda, σ_x e σ_y definem a abertura do envelope Gaussiano ao logo dos eixos x e y, θ é o ângulo, medido no sentido anti-horário, entre a direção de propagação do plano de onda e o eixo x, $x_r=(x-x_0)cos\theta$ e $y_r=(y-y_0)\text{sen}\theta$ definem a posição no plano de onda, ϕ é o deslocamento no plano de onda (assumido ser zero), (x_0,y_0) é a localização de um ponto na imagem original, e (u,v) define a frequência espacial, que pode ser expressa como frequência radial ω e orientação θ, calculadas como

$$\omega = \left(u^2+v^2\right)^{1/2}, \tag{2.6}$$

$$\theta = \arctan\left(\frac{v}{u}\right), \tag{2.7}$$

e

$$\sigma_u = \frac{1}{2\pi\sigma_x}, \qquad \sigma_v = \frac{1}{2\pi\sigma_y}. \tag{2.8}$$

As relações de localização e orientação descritas pelas Equações (2.6), (2.7) e (2.8) do filtro de Gabor podem ser visualizadas mais explicitamente na sua representação no domínio da frequência (ver Figura 2.6).

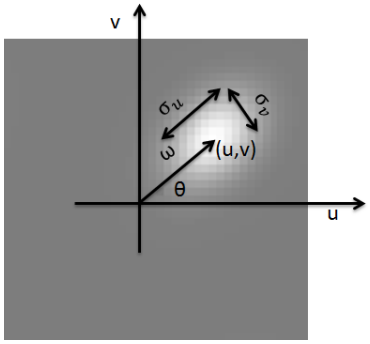


Figura 2.6: Filtro de Gabor 2D no domínio da frequência e as relações entre $\omega = 0,5$ pixel/ciclo e $\theta = \frac{\pi}{4}$.

O princípio da incerteza estabelece que o produto do espalhamento (i.e., a incerteza) de um sinal nos domínios do tempo e da frequência deve exceder ou ser igual a uma constante fixa, ou seja, deve obedecer à relação

$$\Delta t \Delta f \geq c, \tag{2.9}$$

em que c é uma constante, e Δt e Δf representam a medida do espalhamento do sinal nos domínios do tempo e da frequência, respectivamente. A implicação deste princípio é que a precisão com que se mensura um sinal em um domínio limita a precisão da medida em outro domínio. Gabor (1946) demonstrou que o hoje denominado filtro de Gabor complexo, baseado em uma função gaussiana, atende o compromisso de menor valor para c entre a localização nos domínios do tempo e da frequência. É importante salientar esta propriedade do Filtro de Gabor, visto que este filtro é a proposição de uma transformada de Fourier localizada por meio de uma janela, em que a janela é uma Gaussiana. Graças a esta proposição é possível, utilizando um filtro de Gabor, extrair informação localmente, sem considerar o sinal completo para produzir suas respostas.

Para extrair informação útil de uma imagem um conjunto de filtros de Gabor com diferentes frequências e orientações é requerido. Como recomendado por Daugman (Daugman; 1985), empregam-se orientações igualmente dispostas, de acordo com

$$\theta_m = \frac{\mu\pi}{M}, \quad \omega_n = \omega_{max}/\sqrt{2}^n, \tag{2.10}$$

em que $\mu = 0, \ldots, M-1$, com M igual ao número de orientações que se quer variar para os filtros de Gabor, $n = 0, \ldots, N-1$, sendo N o número de frequências, sendo ω_{max} a frequência máxima. De acordo com o teorema da amostragem de Nyquist um sinal contendo frequências maiores que metade da frequência de amostragem não pode ser completamente reconstruído. Então, o limite superior para frequência de uma imagem é $0,5$ $pixel/ciclos$, enquanto que o limite inferior é 0 (Daugman; 1985).

Definido como uma função Gaussiana que age como o envelope de um plano de onda, o filtro de Gabor é capaz de extrair características em uma cena visual através de uma operação de convolução com uma imagem local $\boldsymbol{I}(x_0, y_0)$, ou seja,

$$GR_{\omega_n,\theta_m} = \psi(\mathbf{p}) \otimes \boldsymbol{I}(x_0, y_0), \tag{2.11}$$

em que GR_{ω_n,θ_m} representa o resultado da convolução 2D (respesentada pelo símbolo $\otimes$) de uma resposta de Gabor na orientação θ_m e frequência radial ω_m.

É usual aplicar vários filtros de Gabor com diferentes orientações e frequências, com o intuito de extrair características significativas em uma imagem. Utilizando um banco de filtros, as faces que assumem uma pose não vertical, não frontal ou sob oclusão parcial podem ser bem representadas. Isto se torna possível pelo uso da representação holística das respostas de Gabor da imagem da face. Ainda que esta esteja parcialmente oculta na imagem, é possível detectá-la como face, mesmo sob oclusão parcial as respostas de Gabor realçam parte da face.

É comum utilizar o banco de filtros de Gabor composto por 5 frequências e 8 orientações (Lades et al.; 1993; Shen et al.; 2007; Serrano et al.; 2011). Contudo, muitos autores usam sua própria parametrização. Nesta Tese será adotada uma parametrização própria, assumindo o compromisso de compor um vetor de características representativo, mas que não venha a comprometer o requisito de dinâmica de um sistema de seguimento de face numa sequência de quadros em um vídeo. Assim, propomos a utilização de uma só frequência e 4 orientações. A quantidade de orientações utilizadas para a representação da face com o banco de filtros de Gabor foi definida como quatro por se considerar uma faixa de orientações suficiente para caracterizar as rotações da face nas sequências de vídeo. Empregam-se orientações igualmente dispostas de acordo com $\theta = \frac{\mu\pi}{4}$, em que $\mu = 0, \ldots, M-1$, com M igual a 4. Note-se que o banco de filtros de Gabor utilizado decresce a redundância de informação de $40(5x8)$ para $4(1x4)$.

A frequência máxima utilizada para compor o banco de filtros é $\omega_{max} = 0,25 pixel/ciclos$, por observer que resultados melhores foram conseguidos com este valor. Isto se deve ao fato das características de face necessitarem de uma frequência mais estreita para uma representação eficiente. Pelo fato das respostas de Gabor apresentarem resultados melhores em imagens com maior contraste, é utilizada a banda verde da imagem original. Observe-se que não seria uma boa estratégia extrair as respostas de Gabor de cada uma das três bandas de cor, pois isto aumentaria significativamente a dimensão do vetor de características.

As Figuras 2.7 e 2.9 exibem a parte real do banco de filtros de Gabor para as 4 orientações e a frequência adotadas, nos domínios espaciais 2D e 3D, respectivamentee e sua representação no domínio da frequência é exibido na Figura 2.8. Um exemplo de $GR_{\omega,\theta}$ para uma imagem de face, no caso aquela mostrada na Figura 2.10, é exibido na Figura 2.11.

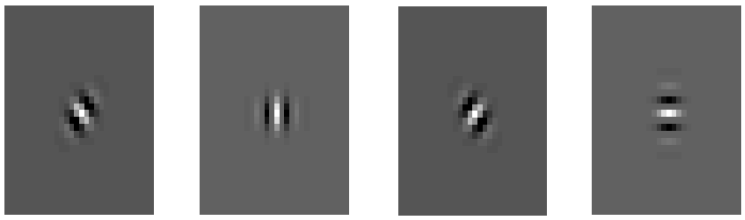


Figura 2.7: Banco de Filtro de Gabor para uma frequência $\omega = 0,25$ pixel/ciclo e 4 orientações $\theta = \frac{\mu\pi}{4}$, em que $\mu = 0,1,2,3$.

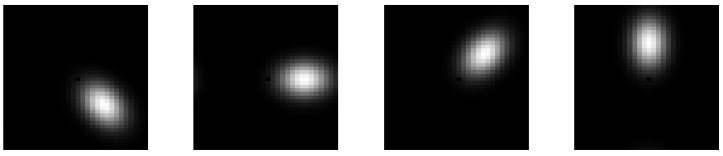


Figura 2.8: Representação no domínio da frequência dda magnitude do Banco de Filtro de Gabor exibido na Figura 2.7.

Figura 2.9: Exibição 3D da parte real do banco de filtro de Gabor exibido na Figura 2.7.

A representação de uma imagem por um banco de filtros que é não ortogonal é altamente redundante, isto é, as respostas dos filtros são altamente correlacionadas.

Figura 2.10: Exemplo de imagem para o padrão face.

Contudo, cabe salientar que cada uma das saídas geradas pela filtro de Gabor traz alguma informação adicional, muitas vezes não presente em outras respostas, e é esse fato que será explorado pela abordagem desenvolvida nesta seção. Em particular, a característica selecionada para representar os padrões face e não face é o valor máximo absoluto das respostas de Gabor, ou seja,

$$MaxGR = \max \left| GR_{\omega_n,\theta_m} \right|, \tag{2.12}$$

para cada ponto (x, y) nas quatro saídas de Gabor é escolhido o maior valor absoluto para compor $MaxGR$. Nesta abordagem são selecionadas as informações redundantes e agregadas as informações adicionais que possam haver de cada resposta. A imagem de face é cortada de modo a adquirir a dimensão de 30×40. Portanto, o vetor de características dos padrões, que adquire a dimensão de cada amostra, então, é 30×40 (Figura 2.12).

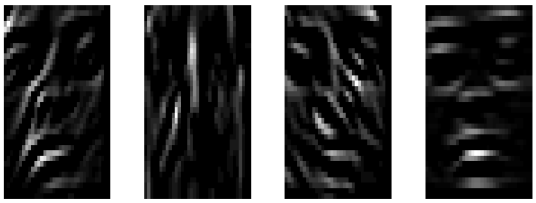


Figura 2.11: Resposta de Gabor para resultado da convolução de uma imagem de face (Figura 2.10) com o banco de filtros de Gabor parametrizado com quatro orientações e uma frequência.

2.4 Máquina de Vetores de Suporte

Máquinas de Vetores de Suporte, ou simplesmente SVM (do inglês *Support Vector Machines*), consistem em uma técnica de aprendizado supervisionado aplicável à classificação e regressão fundamentada na Teoria de Aprendizado Estatístico (Burges; 1998). Uma propriedade importante das máquinas de vetores de suporte é que a determinação

Figura 2.12: Máximo absoluto para as respostas do banco de filtros de Gabor apresentadas na Figura 2.11.

dos parâmetros do modelo corresponde a um problema optimização convexa e assim qualquer solução local é também um ótimo global. O classificador SVM tem como objetivo realizar a separação entre duas classes, de forma que a margem entre a superfície de separação e as amostras mais próximas a essa superfície seja maximizada. Diferentemente do que acontece com a rede neural perceptron treinada por retro-propagação do erro (Bishop; 2006), em que os vetores de peso são influenciados por todas as amostras de treinamento, no SVM a superfície de decisão é definida apenas por alguns elementos do conjunto de treinamento. Esses elementos recebem o nome de **vetores de suporte**.

O problema de detecção de face dentro do problema de abordado neste trabalho é um caso de classificação não separável entre padrões face e não face, especialmente se for levado em consideração que a face a ser detectada não estará no plano de captura da imagem. O padrão não-face compreende a todo o resto do universo. Devido à complexidade do problema se requer uma máquina de decisão com alto poder de generalização, onde se encaixa o SVM, descrito a seguir.

A função de discriminante para o caso de classificação com duas classes é dada por

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b, \tag{2.13}$$

onde ϕ representa uma função de transformação do espaço de características, $\mathbf{w}$ corresponde a uma combinação linear e b corresponde ao bias. O conjunto de treinamento é composto de N vetores de entrada $\mathbf{x}_1, \ldots, \mathbf{x}_N$, com rótulos correspondentes $t_1, ..., t_N$, em que $t_n \in \{-1, 1\}$, e novos pontos $\mathbf{x}$ são classificados de acordo com o sinal de $y(\mathbf{x})$.

Considerando que o conjunto de dados é linearmente separável no espaço de características, então existe pelo menos uma escolha dos parâmetros $\mathbf{w}$ e b tal que a Equação (2.13) satisfaça $y(\mathbf{x}_n) > 0$ para os pontos que possuam $t_n = +1$ e $y(\mathbf{x}_n) < 0$ para os pontos que possuem $t_n = -1$ tal que $t_n y(\mathbf{x}) > 0$ para todos os pontos de treinamento.

No SVM a superfície de decisão é escolhida de forma a maximizar a margem entre essa superfície e o ponto do conjunto de treinamento mais próximo. A Figura 2.13

apresenta o conceito da margem, e mostra como os vetores de suporte atuam na formação da superfície de decisão. A distância de um ponto $\mathbf{x}$ até um hiperplano definido

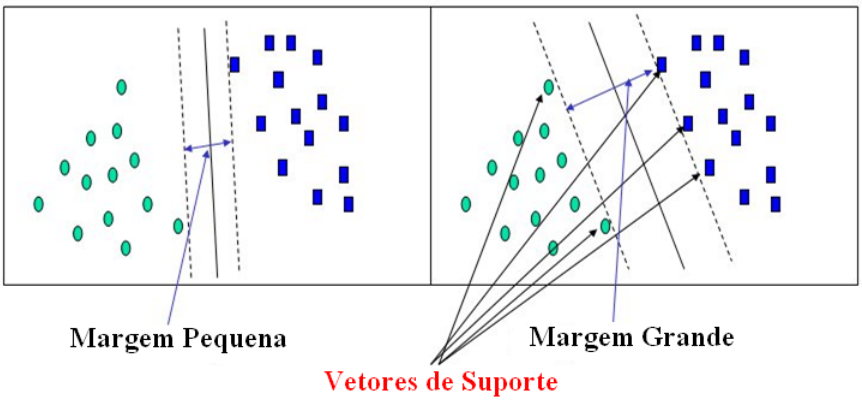


Figura 2.13: Representação da margem. Função dos vetores de suporte, indicados pelos círculos, na formação da superfície de decisão.

por $y(\mathbf{x}) = 0$ (Equação (2.13)) é dada por $|y(\mathbf{x})| / \|\mathbf{w}\|$, em que $\|\cdot\|$ representa a norma Euclidiana. Como se está interessado nas soluções classificadas corretamente, tal que $t_n y(\mathbf{x}_n) > 0$ para todo n, então a distância do ponto $\mathbf{x}_n$ até a superfície de decisão é dada por

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n \left(\mathbf{w}^T \phi(\mathbf{x}) + b\right)}{\|\mathbf{w}\|}. \tag{2.14}$$

Deseja-se maximizar a distância perpendicular entre a fronteira de decisão e o ponto $\mathbf{x}_n$ mais próximo (do conjunto de treinamento), através da otimização de $\mathbf{w}$ e b. Então, a solução que maximiza a margem é encontrada pela solução de

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[t_n \left(\mathbf{w}_T \phi(\mathbf{x}_n) \right) + b \right] \right\}. \tag{2.15}$$

Reescalando por um fator k,$\mathbf{w} \to k\mathbf{w}$ e $b \to kb$, a distância de qualquer ponto $\mathbf{x}_n$ até a superfície de decisão, dada por $t_n y(\mathbf{x}_n) / \|\mathbf{w}\|$, não é alterada. Portanto, pode-se ajustar

$$t_n \left(\mathbf{w}^T \phi(\mathbf{x}_n) + b \right) = 1 \tag{2.16}$$

para o ponto mais próximo da superfície. Assim, todos os pontos vão satisfazer à restrição

$$t_n \left(\mathbf{w}^T \phi(\mathbf{x}_n) + b \right) \geq 1, n = 1, \ldots, N. \tag{2.17}$$

Com essa restrição o problema descrito por (2.15) pode ser simplificado para a maximização de $\|\mathbf{w}\|^{-1}$, o que é equivalente a minimizar $\|\mathbf{w}\|^2$. Ou seja, deve-se resolver o problema de otimização

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2, \tag{2.18}$$

sujeito à restrição dada por (2.17). O fator 1/2 é incluído por conveniência, uma vez que a obtenção de $\mathbf{w}$ e b será feita por derivação (através da derivada da função Lagrangeana, como será visto a seguir) com relação a $\mathbf{w}$ e b.

Para resolver o problema de otimização descrito por (2.18), sujeito às restrições dadas por (2.17), utilizam-se os multiplicadores de Lagrange $a_n \geq 0$, com um multiplicador para cada restrição em (2.17), o que leva à função Lagrangeana

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left\{ t_n \left(\mathbf{w}^T \phi(\mathbf{x}_n) + b \right) - 1 \right\}, \tag{2.19}$$

em que $a = (a_1, \ldots, a_N)^T$. O sinal menos na frente do multiplicador de Lagrange se deve ao fato de estar sendo feita uma minimização com relação a $\mathbf{w}$ e b, e uma maximização com relação a $\mathbf{a}$. Fazendo as derivadas de $L(\mathbf{w}, b, \mathbf{a})$ com relação a $\mathbf{w}$ e b iguais a zero, obtém-se as duas condições

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \tag{2.20}$$

e

$$0 = \sum_{n=1}^{N} a_n t_n. \tag{2.21}$$

Eliminando $\mathbf{w}$ e b de $L(\mathbf{w}, b, \mathbf{a})$ e usando essas condições obtém-se a representação dual do problema de maximização da margem, dada por

$$\tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{2.22}$$

sujeita às restrições

$$a_n \geq 0, \qquad n = 1, \ldots, N, \tag{2.23}$$

e

$$\sum_{n=1}^{N} a_n t_n = 0, \tag{2.24}$$

em que as predições são feitas a partir da combinação linear de uma função *kernel*, definida como $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$, calculada a partir os dados de treinamento.

O conceito de *kernel*, formulado como um produto interno no espaço de características, permite construir o conceito de *kernel trick* ou substituição de *kernel*. A idéia geral deste conceito é que se um algoritmo é formulado de tal forma que o vetor de entrada $\mathbf{x}$ é introduzido na forma de um produto escalar, pode-se substituir este produto escalar por algum *kernel*.

Para classificar uma nova amostra usando o modelo treinado, avalia-se o sinal de $y(\mathbf{x})$ definido pela Equação (2.13). Esse valor pode ser expresso em função dos parâmetros a_n e da função *kernel*, pela substituição de $\mathbf{w}$ usando a Equação (2.20), o que resulta em

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b. \tag{2.25}$$

Esse tipo de otimização deve obedecer às condições de Karush-Kuhn-Tucker (KKT), que nesse caso requer que as três propriedades a seguir sejam válidas (Bishop, 2006)

$$a_n t_n > 0, \tag{2.26}$$

$$t_n y(\mathbf{x}_n) - 1 \geq 0, \tag{2.27}$$

$$a_n \{t_n y(\mathbf{x}_n) - 1\} = 0. \tag{2.28}$$

Então, para todos os pontos do conjunto de treinamento deve-se ter $a_n = 0$ ou $t_n y(\mathbf{x}_n) = 1$. Os pontos para os quais $a_n = 0$ não aparecem no somatório em (2.25), e, portanto não apresentam função na classificação de novos pontos.

Depois de encontrado o vetor $\mathbf{a}$, pode-se encontrar o valor do parâmetro de polarização b, considerando que qualquer um dos vetores de suporte deve satisfazer $t_n y(\mathbf{x}_n) = 1$. Por (2.25) tem-se que

$$t_n \left(\sum_{m \in S} a_m t_m k(x_n, x_m) + b \right) = 1, \tag{2.29}$$

em que S representa o conjunto de índices dos vetores de suporte. Embora o valor de b possa ser encontrado resolvendo essa equação para apenas um vetor de suporte, uma solução mais confiável é encontrada calculando-se o valor de b para todos os vetores de suporte e, posteriormente, dividindo o valor pelo número total de vetores de suporte N_S. Multiplicando a Equação (2.29) por t_n, usando o fato de que $t_n^2 = 1$, e calculando a média, obtém-se

$$b = \frac{1}{N_S} \sum_{n \in S} \left(t_n - \sum_{m \in S} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right), \tag{2.30}$$

em que N_S é o número total de vetores de suporte.

Se for assumido que os dados de treinamento são linearmente separáveis no espaço de características $\phi(\mathbf{x})$, o SVM resultante dará uma separação exata dos dados de treino no espaço de entrada original $\mathbf{x}$, embora a superfície de decisão seja não linear. Na prática, contudo, as distribuições condicionais das classes podem se sobrepor e, nestes casos, a separação exata do conjunto de treinamento pode levar a uma generalização pobre.

É necessário uma maneira de modificar o SVM para permitir que alguns pontos de treinamento sejam mal classificados. A modificação é feita de tal forma que a alguns pontos de dados seja permitido estarem do lado errado da margem fronteira de separação, mas com uma penalidade que aumente com a distância daquela fronteira. Para o problema de otimização é conveniente fazer desta penalidade uma função linear desta distância. Para fazer isto, introduzem-se as variáveis de folga $\xi_n \geq 0$, com $n = 1, \ldots, N$, com uma variável da folga para cada ponto de dados. Estas são definidas por $\xi_n = 0$, para pontos de dados que são ou estão dentro da fronteira da margem correta, e $\xi_n = |t_n - y(\mathbf{x}_n)|$ para outros pontos. Assim, um ponto de dados que está na fronteira de decisão $y_n(x_n) = 0$ terá $\xi_n = 1$, e pontos de dados com $\xi_n > 1$ serão mal classificados. As restrições de classificação exata são substituídas com

$$t_n y(\mathbf{x}_n) \geq 1 - \xi_n, \qquad n = 1, \ldots, N, \tag{2.31}$$

em que as variáveis *slack* são restringidas a satisfazerem $\xi_n \geq 0$. Pontos de dados para os quais $\xi_n = 0$ são corretamente classificados e estão na margem ou estão do lado correto da margem. Pontos para os quais $0 < \xi_n \leq 1$ estão na margem, porém do lado correto da fronteira de decisão e aqueles pontos para os quais $\xi_n > 1$ estão no lado errado da fronteira de decisão e são mal classificados. Isto é algumas vezes descrito como relaxação da restrição de margem rígida para dar uma margem suave e permitir que alguns pontos do conjunto de dados sejam mal classificados. Nota-se que enquanto variáveis de folga permitem que alguns distribuições de classes se sobreponham, este método ainda é sensível aos *outliers*, devido à penalidade para má classificação aumentar linearmente com ξ.

A meta é maximizar a margem enquanto penaliza-se suavemente os pontos que fiquem do lado errado da fronteira da margem. Portanto, minimiza-se

$$C \sum_{n=1}^{N} \xi_n + \frac{1}{2} \|\mathbf{w}\|^2, \tag{2.32}$$

em que o parâmetro $C > 0$ controla o compromisso entre a penalidade da variável de folga e a margem. Já que qualquer ponto que é mal classificado tem $\xi_n \geq 1$, segue que

$\sum_n \xi_n$ é o limite superior sobre o número de pontos mal classificados. O parâmetro C é, portanto, análogo a um coeficiente de regularização, pois controla o compromisso entre minimização do erro de treinamento e controle da complexidade do modelo. No limite $C \to \infty$ recuperar-se-á a primeira máquina de vetor de suporte para dados separáveis.

Deseja-se minimizar 2.32 sujeito às restrições 2.31 juntamento com $\xi \geq 0$. O Lagrangiano correspondente é dado por

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N} a_n \{t_n y(\mathbf{x}_n) - 1 + \xi_n\} - \sum_{n=1}^{N}\mu_n\xi_n \tag{2.33}$$

onde $\{a_n \geq 0\}$ e $\{\mu_n \geq 0\}$ são multiplicadores de Lagrange. O conjunto correspondente de condições KKT são dados por

$$a_n \geq 0 \tag{2.34}$$
$$t_n y(\mathbf{x}_n) - 1 + \xi_n \geq 0 \tag{2.35}$$
$$a_n(t_n y(\mathbf{x}_n) - 1 + \xi_n) = 0 \tag{2.36}$$
$$vk\mu_n \geq 0 \tag{2.37}$$
$$\xi_n \geq 0 \tag{2.38}$$
$$\mu n \xi_n = 0 \tag{2.39}$$

onde $n = 1,\ldots,N$.

Agora otimizar $\mathbf{w}$,b e $\{\xi_n\}$ fazendo uso da definição (2.13) de $y(\mathbf{x})$ para dar

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \tag{2.40}$$
$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad \sum_{n=1}^{N} a_n t_n = 0 \tag{2.41}$$
$$\frac{\partial L}{\partial \xi_n} = 0 \quad \Rightarrow \quad a_n = C - \mu_n. \tag{2.42}$$

Utilizando estes resultados para eliminar $\mathbf{w}$,b e $\{\xi_n\}$ do Lagrangiano, ontem-se o Lagrangiano dual na forma

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{2.43}$$

que é idêntico ao caso separável, exceto pelas restrições que são um pouco diferentes. Para saber quais são estas restrições, oberve que $a_n \geq 0$ é requerido pois são os multiplicadores de Lagrange. Além disto, (2.41) juntamente com $\mu_n \geq 0$ implica $a_n \leq C$. Tem-se, portanto, para minimizar (2.43) com respeita às variáveis duais $\{a_n\}$ sujeito a

$$0 \leq a_n \leq C \tag{2.44}$$

$$\sum_{n=1}^{N} a_n t_n = 0 \tag{2.45}$$

para $n = 1,\ldots,N$, onde (2.44) são conhecidos como retrições de caixa. Isto novamente representa um problema de programação quadrática. Se (2.41) for substituido em (2.13), Vê-se que predições para novos pontos de dados são feitos novamente utilizando (2.25).

Pode-se interpretar a solução resultante. Como antes, um subconjunto de pontos de dados pode ter $a_n = 0$, em cujo caso são contribuem para para o modelo preditivo (2.25). O restante dos pontos de dados constituem os vetores de suporte. Estes possuem $a_n > 0$ e portanto, de (2.37) devem satisfazer

$$t_n y(\mathbf{x}_n) = 1 - \xi_n. \tag{2.46}$$

Se $a_n < C$, então (2.42) implica que $\mu_n > 0$, que, de (2.39) requer $\xi_n = 0$ e daí tais pontos permanecem na margem. Pontos com $a_n = C$ podem estar dentro da margem e podem ser corretamente classificados se $\xi_n \leq 1$ ou classificados erroneamente se $\xi_n > 1$.

Para determinar o parâmetro b em (2.13), vê-e que aqueles vetores de suporte para os quais $0 < a_n < C$ têm $\xi_n = 0$ tal que $t_n y(\mathbf{x}_n) = 1$ e então satifará

$$t_n \left(\sum_{m \in S} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) + b \right) = 1. \tag{2.47}$$

Mais uma vez, uma solução numérica estával é obtida por uma média para dar

$$b = \frac{1}{N\mathcal{M}} \sum_{m \in \mathcal{M}} \left(t_n - \sum_{m \in S} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right). \tag{2.48}$$

onde $\mathcal{M}$ denota o conjunto de índices de pontos de dados tenho $0 < a_n < C$.

Uma alternativa, formulação equivalente à máquna de vetores de suporte, conhecida como ν-SVM, proposto por Schölkopf (Schölkopf et al.; 2000). Isto envolve maximização de

$$\tilde{L}(\mathbf{a}) = -\frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{2.49}$$

sujeito às restrições

$$0 \leq a_n \leq 1/N \tag{2.50}$$

$$\sum_{n=1}^{N} a_n t_n = 0 \tag{2.51}$$

$$\sum_{n=1}^{N} a_n \geq \nu \tag{2.52}$$

Este método tem a vantagem de que o parâmetro ν, que substitui C, pode ser interpretado como limite superior da fração de erros de margem e limite inferior da fração de vetores de suporte.

É importante se ter algoritmos eficientes para resolver o problema da programação quadrática. Primeiro observa-se a função $\tilde{L}(\mathbf{a})$ dada por (2.22) ou (2.43) é quadrática e qualquer ótimo local também será um ótimo global desde que as restrições definem uma região convexa. A solução direta do problema de programação quadrática utilizando técnicas tradicionais é muitas vezes inviável devido ao custo computacional e requerimentos de memória e algumas abordagens mais práticas precisam ser encontradas.

A técnica de *chunking* (Vapnik; 1995) explora o fato de que o valor do Lagrangeano é imutável se se removem as linhas e colunas da matriz *kernel* correspondentes aos multiplicadores de Lagrange que tenham valor zero. Isto permite que o problema completo de programação quadrática seja dividido em uma série de outros problemas menores, cujo objetivo é, eventualmente, identificar todos os multiplicadores de Lagrange nulos e descartar os outros.

Os gradientes conjugados protegidos de Burges (1998) reduz o tamanho da matriz na função quadrática do número de pontos de dados para, aproximadadmente, o número de multiplicadores de Lagrande ao quadrado, porém, isto pode requerer montante execivo de memória para aplicações em uma grande escala. Os métodos de decomposição (Osuna et al.; 1997) também podem resolver uma série de problemas de programação quadrática menores mas são projetados tal que cada um destes seja de um tamanho fixo e por isso a técnica pode ser aplicada a conjuntos de dados arbitrariamente grandes. Contudo, ainda envolve solução numérica de sub problemas de programação quadrática e estes podem ser problemáticos e caros.

Um dos mais populares métodos para treinar máquinas de vetores de suporte é denominada Otimização Mínima Sequencial ou SMO (do inglês *sequential minimal optimization*) (Platt; 1999). Esta técnica toma o conceito de *chunking* ao extremo e considera apenas dois multiplicadores de Lagrange por vez. Neste caso, o subproblema pode ser resolvido analiticamente, evitando a programação quadrática numérica completamente. Heurísticas são dadas para escolher o par de multiplicadores de Lagrange a serem considerados a cada passo. Na prática, o SMO tem um dimensionamento com o número de pontos de dados que é algo entre linear e quadrático, dependendo da aplicação em particular.

Observa-se que funções de *kernel* correspondem a produtos internos nos espaços de característica que, podendo ter alta dimensionalidade ou mesmo infinita. Ao trabalhar

diretamente em termos da função *kernel*, sem introduzir o espaço de característica explicitamente, pode, portanto, parecer que as máquinas de vetores de suporte de alguma forma conseguem evitar a maldição da dimensionalidade. Este não é o caso, no entanto, porque existem restrições entre os valores das características que limitam a dimensão efetiva do espaço de características.

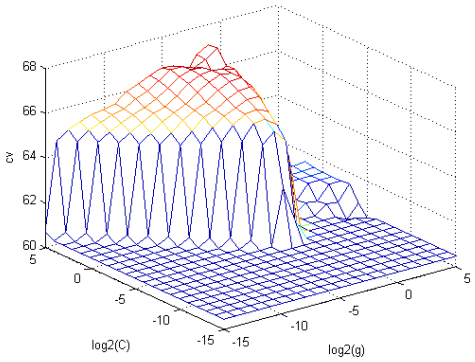


Figura 2.14: *Cross validation.* Valores obtidos durante o procedimento de validação cruzada utilizado para escolha os melhores valores do *kernel* RBF.

O SVM utilizado para detecção de face neste trabalho de doutorado está implementado na biblioteca LibSVM (Chang and Lin; 2011), que utiliza o algoritmo SMO para treinamento com *kernel* RBF, escolhido por ter menor número de parâmetros a serem inferidos, além de apresentar resultados confiáveis na literatura. O kernel RBF é definido pela equação

$$k\left(\mathbf{x},\mathbf{x}'\right)=\exp\left(-\gamma\left\|x-\mathbf{x}'\right\|^2\right). \tag{2.53}$$

Os parâmetros do *kernel* foram escolhidos através de validação cruzada (Bishop; 2006) utilizando o *v-fold* para o grupo de treinamento. Portanto, o grupo de treinamento é dividido em v subgrupos e a cada iteração um dos subgrupos é testado utilizando os dados do classificador treinado com os outros $v-1$ subgrupos. O valor de v é 5 para a validação cruzada desta tese. Essa validação cruzada é necessária para minimizar o problema de sobre ajuste, onde o classificador consegue separar bem as classes apresentadas no treinamento, mas não é capaz de lidar com a generalização.

Os melhores parâmetros γ e C foram encontrados atravez de uma busca exaustivas numa faixa de valores no intervalo $[-5,5]$, com um passo de busca de $0,1$. Uma vez localizado o ponto que apresenta melhor desempenho, uma nova busca é iniciada centrada nesse ponto com um passo de busca em menor escala. Na Figura 2.14 é exibido um gráfico com os valores $\log_2(\gamma)$, $\log_2(C)$ e CV obtidos durante o processo de validação cruzada. O valor CV representa a acurácia obtida durante a validação

cruzada. O valor dos parâmetros para treinar o SVM RBF implementado são $\gamma = 0,5$ e penalidade $C = 16$.

2.5 Filtro de Kalman

A solução clássica para o seguimento dinâmico de face em sequência de vídeo consiste em utilizar um detector de face e um filtro recursivo para estimar a posição do alvo no próximo quadro da sequência (Foytik et al.; 2011; Fu and Han; 2012; Yin et al.; 2011; Karavasilis et al.; 2011; Faux and Luthon; 2012). De acordo com este esquema, o seguimento da face inicia após o alvo ter sido detectado. A localização da posição da face detectada é utilizada como observação para o filtro por toda a sequência de vídeo. O rastreio é feito utilizando um modelo de movimento sem que a detecção do alvo seja aplicada quadro a quadro, por uma questão de menor esforço computacional.

Em caso de oclusão total ou parcial, no esquema clássico, a posição para o próximo quadro é estimada a partir da localização da face no quadro anterior. Quando uma nova observação, a localização da face redetectada, se torna acessível o filtro de kalman estima a posição da face para o quadro seguinte. O seguimento é feito alternando entre estimação de posição e, se necessário, redetecção do alvo.

O seguimento de face permanece um assunto chave em visão computacional. O assunto se torna interessante porque é preciso contornar alguns desafios comumente presentes em cenas adquiridas em ambientes não controlados, como oclusão, modificação da aparência da face, iluminação não uniforme, bem como plano de fundo complexo, movimento arbitrário da face, etc., que contribuem para definir o caráter não estacionário do problema.

Na tarefa de seguir uma face, abordada nesta Tese, pretende-se obter a localização de uma face numa sequência de vídeo adquirida em ambiente não controlado. Esta localização ruidosa se torna acessível após um classificador do tipo SVM RBF varrer uma janela de busca procurando por candidatas a face e encontrando. Então, é necessário estimar onde a face vai estar no próximo quadro da sequência de vídeo, assim prosseguindo ao longo da referida sequência de vídeo.

Para chegar-se à escolha do filtro de Kalman foram feitos testes com o filtro de Kalman discreto e um filtro de partículas MCMC (do inglês *Markov Chain Monte Carlo*) (Chen; 2003), método este que também é utilizado para seguimento de alvos (Hotta; 2009; Lee et al.; 2012). Diferentemente do filtro de Kalman, que é um filtro linear, o filtro de partículas é concebido para resolver a estimação de dados de forma

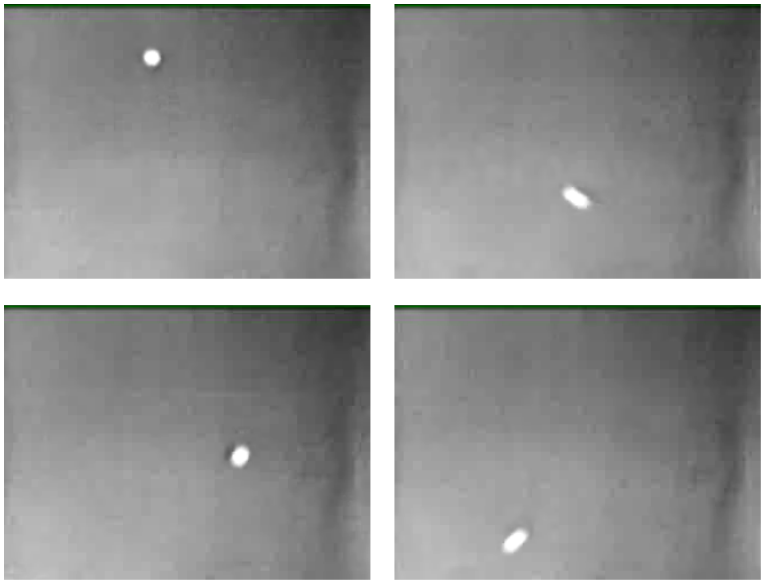


Figura 2.15: Quadros da sequência de vídeo de teste feito com uma esfera sobre um plano de fundo preto.

não linear. Além disto, sob perspectiva estatística, o filtro de Kalman assume que os dados são Gaussianos, enquanto o filtro de partículas é uma solução que admite que não é necessário conhecimento da distribuição dos dados (Arulampalam et al.; 2002). Diante disto, este último também poderia ser adequado para o tema abordado nesta Tese, se fosse assumido que a face a ser seguida pode adquirir movimento e velocidade arbitrários (entretanto, este não é o caso aqui considerado).

Com o intuito de explorar o potencial destes dois filtros, foram examinados os seus desempenhos no seguimento simulado de um alvo com movimento arbitrário. Uma sequência de vídeo foi capturada, na qual o plano de fundo era uma cartolina preta, e uma esfera, de cor cinza claro, com diâmetro de aproximadamente 2 centímetros, foi movida em direção arbitrária e velocidade não constante. O plano de fundo foi escolhido para que não houvesse oclusões. Neste teste o alvo foi segmentado do plano de fundo por uma operação de diferença do plano de fundo com cada quadro. Desta forma foi possível obter a localização real do alvo a cada *frame*, para comparação com as trajetórias descritas quando se utilizavam os dois filtros sob teste.

O resultado do seguimento com o filtro de Kalman discreto apresentou uma trajetória mais próxima da trajetória simulada com a esfera do que o resultado de seguimento obtido com o filtro de partículas. Levando-se em consideração que uma face em uma sequência de vídeo se move menos brusca e rapidamente do que se movia a esfera nas sequências de vídeo de teste, o filtro de partículas não foi adotado na metodologia apresentada nesta Tese, adotando-se o filtro de Kalman discreto. A Figura 2.15 mostra

alguns quadros do teste feito com a esfera se movimentando brusca e rapidamente.

Este resultado é coerente com a pesquisa desenvolvida em (Lee et al.; 2012) que mostra que Kalman apresenta melhor resultado para seguimento de posição de alvo. No trabalho de Lee e colegas (Lee et al.; 2012) são avaliados vídeos simulados com velocidade constante, direção arbitrária, oclusão parcial e total. Os testes apresentam um seguimento melhor desempenhando pelo filtro de Kalman em relação ao filtro de partículas.

O *framework* apresentado nesta tese, diferente do esquema clássico de seguimento de face, apresenta a possibilidade de seguir a face na sequência de vídeo ainda que, na etapa de redetecção, uma nova face não seja encontrada. Ocasiões em que a observação não esté acessível como entrada para o filtro recursivo. A redetecção no *framework* aqui proposto é utilizada a cada quadro, o que pode diminuir a possibilidade de seguimento de falsos positivos. Ainda assim, é possível que o detector falhe e nenhuma face seja detectada na região de busca. O que leva à necessidade de utilizar um filtro recursivo robusto a ausência de obseração, como será aprensentado a seguir.

Muitos problemas requerem a estimação do estado de um sistema que muda com o tempo, como é o caso do seguimento de face, utilizando uma sequência de medições ruidosas feitas sobre o sistema. As equações a diferenças do modelo no espaço de estados são utilizadas para modelar a evolução de sistema no tempo, e medições são feitas assumindo estarem acessíveis no tempo discreto. O método do espaço de estados para modelagem de séries temporais volta sua atenção para o vetor de estados do sistema. O vetor de medições representa observações ruidosas que são relacionadas ao vetor de estados. Com o intuito de analisar e fazer inferência a respeito de um sistema dinâmico, pelo menos dois modelos são requeridos: primeiro, o modelo descrevendo a evolução do estado ao longo do tempo (o modelo do sistema), e, segundo, o modelo relacionando as medições ruidosas aos estados.

O filtro de Kalman permite estimar o estado $\mathbf{x}_k$ de um sistema não estacionário a partir de observações ruidosas $\mathbf{z}_k$, que tenham sua relação com tal estado conhecida, de acordo com as equações

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_{k-1} \tag{2.54}$$

e

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{n}_k, \tag{2.55}$$

em que $\mathbf{v}_{k-1}$ é a sequência de observações ruidosas do processo, e $\mathbf{z}_{k-1}$ a sequência ruidosa de medições, com $k \in N$. $\mathbf{F}_k$ e $\mathbf{H}_k$ são matrizes conhecidas, definindo as funções lineares. As covariâncias de $\mathbf{v}_{k-1}$ e $\mathbf{n}_k$ são, respectivamente, $\mathbf{Q}_{k-1}$ e $\mathbf{R}_k$.

Aqui é considerado o caso em que $\mathbf{v}_{k-1}$ e $\mathbf{n}_k$ tem média zero e são estatisticamente independentes. Observe-se que às matrizes do sistema e de medição, $\mathbf{F}_k$ e $\mathbf{H}_k$, bem como aos parâmetros do ruído, $\mathbf{Q}_{k-1}$ e $\mathbf{R}_k$, é permitido serem variantes no tempo.

A implementação do filtro de Kalman discreto é realizada com um passo de estimação do estado, ou seja,

$$\mathbf{x}_{k|k-1} = \mathbf{F}_k \mathbf{x}_{k-1|k-1} \tag{2.56}$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q}_{k-1} + \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^H, \tag{2.57}$$

e um passo de correção do estado estimado, ou seja,

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}_k \mathbf{z}_k \mathbf{x}_{k|k-1} \right) \tag{2.58}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1}, \tag{2.59}$$

em que $\mathbf{P}_{k|k-1}$ é a covariância do erro de estimação de $\mathbf{x}_k$ dadas as obervações até o instante de tempo $k-1$, e $\mathbf{P}_{k|k}$ é a covariância do erro de estimação de $\mathbf{x}_k$ dadas as observações até o instante de tempo k. Por sua vez,

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^H + \mathbf{R}_k \tag{2.60}$$

e

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^H S_k^{-1} \tag{2.61}$$

são a covariância do termo de inovação $\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_{k|k-1}$ e o ganho de Kalman, respectivamente. Nas equações acima, a matriz hermitiana de $\mathbf{M}$ é denotada por $\mathbf{M}^H$, que é o conjugado complexo de $\mathbf{M}$ transposto.

O objetivo do seguimento é estimar recursivamente $\mathbf{x}_k$ a partir das medições $\mathbf{z}_k$, que são observações que se tornam acessíveis no passo de tempo k. Portanto, o filtro de Kalman é uma solução adequada para o problema apresentado nesta Tese. Usando o filtro de Kalman a localização posterior da face $\mathbf{x}_k$, no próximo quadro da sequência de vídeo, é predita a partir da informação atual (a observação para o filtro de Kalman é a localização da face detectada pelo SVM no quadro atual).

Para o problema proposto nesta Tese, em cada instante de tempo k é assumido que a face está se movendo com velocidade constante. O modelo de movimento da face utilizado no filtro de Kalman para seguir a face tem matriz de transição de estados $\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ e matriz de medições $\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$.

Em um seguimento de face, é possível que, em alguns momentos, a face não seja detectada, ou por estar sob oclusão ou por não ter sido corretamente detectada pelo classificador. Nestes dois casos, o SVM não detecta uma face. Nestas situações não existe qualquer observação $\mathbf{z}_k$ da localização da face no tempo k. O problema do filtro de Kalman na ausência de observação, é resolvido utilizando o seguinte teorema apresentado em (Cipra and Romera; 1997) que perminte dimensões variáveis para o modelo (2.54) - (2.55).

Teorema. Considere o modelo (2.54) - (2.55) em que as dimensões dos vetores $\mathbf{x}_k$ e $\mathbf{y}_k$ podem variar com o tempo. Então, a formulação (2.56) - (2.59) permanece válida. Se o vetor de observações $\mathbf{y}_k$ está ausente no tempo k, então (2.58) e (2.59) devem ser substituidos por

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} \tag{2.62}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} \tag{2.63}$$

A estratégia proposta em (Cipra and Romera; 1997), descrita anteriormente, se mostra adequada nos testes explanados no Capítulo 5, e por isto é aqui utilizada. Os rastreamentos de face clássicos que utilizam filtro de Kalman, o filtro estima a localização da face ao longo da sequência de vídeo a partir de uma posição observada anteriormente. A aborgadem adotada nesta tese dá a possibilidade de estimar a posição da face ainda que não se obtenha uma medição de posição em quadros anteriores.

Os experimentos a respeito do desempenho do Seguidor Dinâmico com Vetores de Suporte são mostrados no Capítulo 4, onde são analisados e discutidos os resultados da utilização do *framework* aqui proposto em vários vídeos selecionados para os testes de validação do sistema.

Capítulo 3

O Seguidor Dinâmico com Vetores de Suporte - SDVS

Neste capítulo a maneira como as técnicas descritas no Capítulo 2 são integradas e cooperam entre si na resolução do problema em questão, a saber, detecção e seguimento de faces em sequências de vídeo, é apresentada.

3.1 Detectando e Seguindo uma Face

O problema abordado nesta Tese está inserido como um componente essencial no fluxograma de processamento de um sistema de reconhecimento de indivíduos (Li and Jain; 2005), como pode ser observado na Figura 3.1, a saber, o módulo de detecção e seguimento de face. Para se compreender o papel do módulo de detecção e seguimento dentro do sistema de reconhecimento de face, são descritos a seguir, de forma resumida, os procedimentos que são realizados em cada um dos módulos do referido fluxograma:

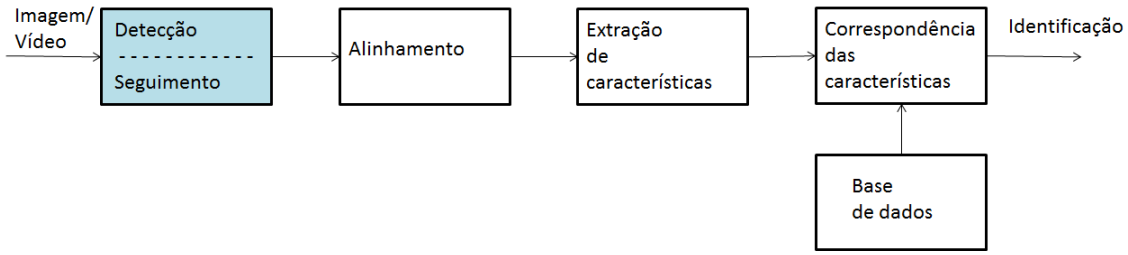


Figura 3.1: Fluxo de processamento para detecção e reconhecimento de face modificado de (Li and Jain; 2005).

- a detecção de face segmenta a face do plano de fundo na imagem. No caso de sequências de vídeo, a face detectada precisa ser seguida por muitos quadros através de um componente de seguimento de face. Enquanto a detecção de face provê uma estimativa da localização da face, características faciais (por exemplo, olhos, nariz, boca, esboço da face) são localizadas. Isto pode ser feito utilizando um módulo de detecção de pontos marcantes da face, ou por um módulo de alinhamento de face;

- no módulo alinhamento da face é realizado um procedimento para normalizar a face, geométrica e fotometricamente. Isto é necessário pois se espera que os métodos de reconhecimento sejam capazes de reconhecer imagens de face com variação de pose e iluminação. O processo de normalização geométrica transforma a face em retângulo, após cortar a face. Já a processo de normalização fotométrica normaliza a face a partir de propriedades como iluminação ou escala de cinza;

- a extração de características da face é realizada na face normalizada, para extrair informações relevantes que sejam úteis para distinguir faces de diferentes pessoas, e é necessário que as informações extraídas sejam robustas a variações geométricas e fotométricas;

- no módulo correspondência de faces as características extraídas de uma face de entrada são comparadas contra uma ou muitas das faces que estão armazenadas em uma base de dados. A saída do módulo correspondência é *sim* ou *não* para verificação, para correspondência de 1:1 ou não, respectivamente. Já para identificação da pessoa a partir da face, em que é verificado se a face corresponde a uma das N faces armazenadas em uma base de dados, a saída é a *identidade* da face de entrada, quando a medida de correspondência encontrada permite o reconhecimento com segurança, ou *desconhecido*, quando o escore de correspondência está abaixo de um limiar pré-estabelecido.

Pode-se perceber, então, a alta dependência destes sistemas sobre cada um dos módulos anteriores, em especial sobre o módulo de detecção e seguimento, mostrando assim o quanto é relevante dispor de um módulo adequado para realizar a detecção e o seguimento de face, que é o problema abordado nesta Tese de Doutorado.

Para tratar tal problema, esta Tese propõe uma abordagem para detectar e seguir uma face em vídeos coloridos capturados em ambientes reais, ou seja, vídeos coloridos sem restrições de iluminação e/ou movimento do alvo. É proposto, então, o *framework* Seguidor Dinâmico com Vetores de Suporte - SDVS, como solução para o módulo de detecção e seguimento de face, o qual se apresenta de forma diferenciada dentro do

fluxo de processamento apresentado anteriormente (Figura 3.1), conforme resumido a seguir:

- a detecção de face se dá pela detecção da informação completa da face, o que na literatura é conhecido como método holístico, ou método baseado na aparência. São extraídas respostas de Gabor e é seguido a localização do centro de massa do pedaço da imagem detectada como face;
- as características de Gabor, selecionadas para compor o vetor de características representativo da face, são robustas quanto a iluminação, a variações de escala, a rotação e a translação. A escolha das características de Gabor se mostra adequada, por se tratar uma ferramenta matemática capaz de distinguir muito bem faces de outros padrões de objetos, assim como distinguir faces de indivíduos diferentes. Isto confere ao sistema a possibilidade de não necessitar escolher uma nova ferramenta matemática para o reconhecimento de face. Por fim, o filtro de Gabor vem sendo, recentemente, bastante utilizado no reconhecimento de face (Yang et al.; 2012);
- antes de extrair as características de Gabor, o SDVS faz uso de um algoritmo de compensação de iluminação, utilizado para contornar o problema da iluminação não uniforme em ambientes não controlados. Desta forma, as regiões de pele podem ser bem identificadas, reduzindo-se assim o esforço computacional na busca por face.

A estratégia de compensação de iluminação e a seleção de características de Gabor utilizadas para detecção de face, propostas como parte do sistema SDVS, altera de forma significativa o módulo de detecção e seguimento de face do fluxograma da Figura 3.1, de forma que é possível obter um novo esquema de processamento para reconhecimento de face: os módulos de alinhamento e de extração de características são incorporados ao módulo de detecção e seguimento aqui proposto, como pode ser visto na Figura 3.2. Neste trabalho não é feito alinhamento, já que a face pode ser detectada sob diferentes orientações, pela utilização das respostas de Gabor. O *framework* proposto nesta tese detecta e segue face. A etapa de reconhecimento de indivíduos é incluído nos trabalhos futuros.

É preciso salientar que nesta Tese não se pretende apresentar um sistema de reconhecimento de face em vídeo. Pretende-se, com esta explanação, apenas mostrar a possibilidade diferenciada ao se utilizar a abordagem de detecção e seguimento de face proposta nesta Tese nos sistemas de reconhecimento em vídeo.

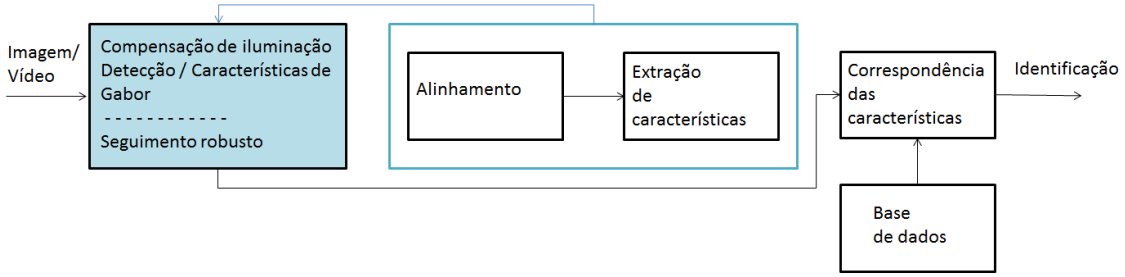


Figura 3.2: Fluxograma de um sistema de reconhecimento de face em vídeo, utilizando a abordagem proposta nesta Tese para seguimento de face.

3.2 O *Framework* SDVS

Em um sequência de vídeo colorido composta por T quadros, o quadro I_t, com $t = \{1, \ldots, T\}$, de dimensão 240×320 pixels, é submetido ao algoritmo de compensação de iluminação aqui proposto, baseado naquele proposto em (Chen and Grecos; 2005). O resultado é uma imagem compensada $I_t^{(\text{comp})}$, sobre a qual serão identificadas regiões de pele, através de limiares empregados em cada um dos três canais do espaço de cores RGB, de forma que

$$Ipele_t = Limiares\left(I_t^{(\text{comp})}\right), \tag{3.1}$$

onde $Ipele_t$ é uma imagem binária em que os pixels identificados como sendo da cor de pele recebem o valor 1, enquanto os demais recebem o valor 0, via a aplicação dos *Limiares*, como descrito na Seção 2.2. A imagem $Ipele_t$, de dimensão 240×320 pixels, constitui um guia para a busca de candidatas a face, já que a procura por face é realizada apenas nas regiões da imagem $Ipele_t$ em que o valor do pixel é 1 (para redução do custo computacional). Enquanto não houver sido detectada a face a ser seguida no vídeo, a janela de busca por face é toda a região de pixels com valor 1 em $Ipele$. Isto acontece quando t é igual a 1 (o primeiro quadro do vídeo) ou quando $t > 1$, neste caso se nenhuma face tiver sido detectada nos quadros até $t-1$. Nos casos em que há uma face sendo seguida, a região de busca por face é reduzida para uma janela de 60×80 pixels, centralizada na posição (x, y) estimada por um filtro de Kalman, janela esta em que se prevê que deve haver região da imagem identificada como pele.

A busca por candidatas a face nas regiões de pele é realizada no canal verde da imagem $Icompensada_t$, e se dá através de uma varredura nestas regiões de pele com uma janela de 30×40 pixels. Cada pedaço $Icandidata$ do canal verde de $Icompensada_t$, varrido na busca por face, é apresentado a um classificador SVM RBF. As características a serem apresentadas a um classificador do tipo SVM precisam ser apresentadas na forma de um vetor. Assim, a resposta máxima absoluta do banco de filtros de Gabor, que compreende as características selecionadas para representar os padrões face e não

face, são vetorizadas (seus pixels são agrupados em um vetor de dimensão $1200x1$), como ilustrado na Figura 3.3.

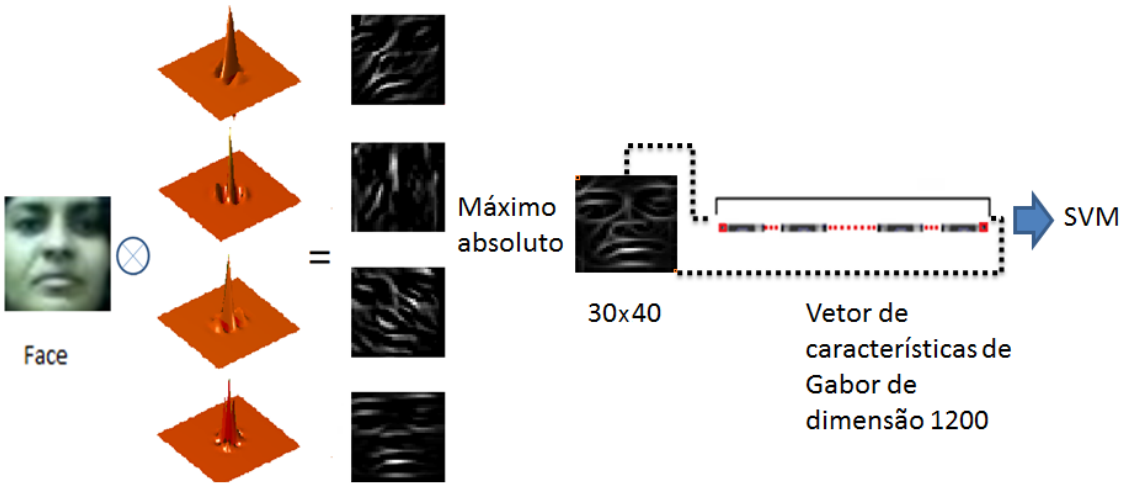


Figura 3.3: Característica selecionada para treinar o classificador SVM: máximo absoluto das respostas do banco de filtros de Gabor, em forma vetorizada.

Assim, a *Icandidata* detectada como face pelo SVM RBF é o alvo a ser seguido pelos próximos quadros que compõem o vídeo colorido.

O desafio de seguir uma face sob condições de um ambiente não controlado pode ser resolvido utilizando um vetor de características representativo, robusto às restrições de rotação, translação, iluminação e oclusão, associado a um filtro de Kalman discreto, responsável por estimar a posição da face no quadro subsequente àquele analisado no instante t, solução adotada no *framework* SDVS.

A localização (x, y) do centro da *Icandidata* detectada como face no quadro t pelo SVM corresponde a uma observação que se torna acessível para que o filtro de Kalman discreto, descrito na Seção 2.5, estime e corrija a localização da face para o próximo quadro do vídeo. Nos casos em que nenhuma face foi detectada, e assim não há observação acessível, o filtro de Kalman estima a posição da face no próximo quadro atualizando a localização da face detectada no quadro anterior. Esta é uma estratégia importante, pois permite que seja estimada a localização de uma face que eventualmente não tenha sido detectada pelo SVM. Sua utilização, como é feito no SDVS, permite reduzir a região de procura por face, que permanece sendo uma janela em torno da posição atualizada pelo filtro de Kalman, contribuindo significativamente para reduzir o custo computacional envolvido no processo (que seria muito maior caso a procura por face fosse feita em toda a área reconhecida como da cor de pele). Note-se, então, que o filtro de Kalman atua de duas formas distintas: estimando a posição do centro da área

correspondente à face detectada no quadro t da sequência de vídeo, ou atualizando a posição desse mesmo centro, a partir do valor encontrado no quadro $t-1$ da sequência de vídeo. Episódios de observação não acessível podem ocorrer por causa de um falso negativo devolvido pelo SVM, ou por algum tipo de oclusão da face existente naquele quadro do vídeo. Neste último caso, o SVM só é capaz de retornar falso verdadeiro.

É importante ressaltar que em cada quadro a região de busca de face, delimitada de acordo com o caso, como descrito anteriormente, é varrida usando-se o SVM, à procura da ocorrência de uma face. Esta estratégia evita que um falso positivo, detectado pelo SVM em algum instante, seja seguido por muitos quadros. Desta forma, a cada quadro uma $Icandidata$ que tenha sido anteriormente computada como falso positivo gera uma janela de busca no próximo quadro, em que se utiliza novamente o SVM, e o resultado desta nova busca por face pode ser um verdadeiro negativo, ou seja, o SDVS permite que um falso positivo possa ser perdido ao longo de um ou mais dos quadros seguintes àquele em que ele foi gerado, conferindo ao SDVS características de robustez a falsos positivos (capacidade de recuperação da localização de face).

Um exemplo ilustrativo da utilização do SDVS em uma sequência de vídeo é ilustrado a seguir, onde se detalham os passos descritos usando alguns dos quadros de uma das sequências de vídeo usadas para teste do *framework*:

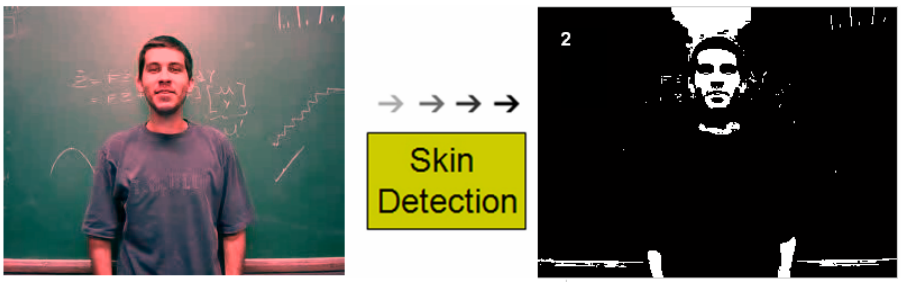


Figura 3.4: Regiões de cor de pele e detecção de face.

- passo 1: no primeiro quadro da sequência de vídeo, pelo fato de uma estimativa prévia de posição da face não estar disponível, a face é procurada exaustivamente em todas as regiões da imagem identificadas como cor de pele. A face assim detectada se torna a observação atual para o filtro de Kalman. Esta observação é obtida utilizando-se o SVM RBF em cada região de cor de pele, como é exemplificado na Figura 3.4;

- passo 2: a estimativa da localização da face para o próximo quadro é, então, obtida pelo filtro de Kalman, a partir das observações disponíveis, como ilustrado na Figura 3.5;

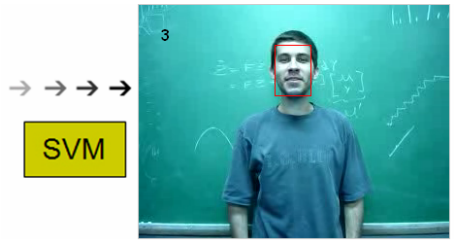


Figura 3.5: Posição da próxima posição do alvo, estimada pelo filtro de Kalman.

- passo 3: uma nova observação é alcançada no ponto estimado no passo anterior (Figura 3.6). Se uma nova observação não se torna acessível, é realizada uma busca numa janela de cor de pele numa vizinhança centralizada na posição estimada utilizando o SVM RBF novamente. A vizinhança de busca, neste passo, é limitada por uma janela de $80x60$ pixels, dimensão esta adotada a partir de testes;

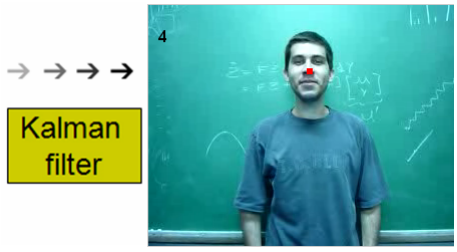


Figura 3.6: Nova face é detectada na janela de busca.

- passo 4: se o alvo é detectado na região de interesse, o algoritmo retorna ao passo 2. No entanto, se nenhum alvo é detectado na região, o algoritmo retorna ao passo 1, tentando conseguir uma nova observação inicial.

A Figura 3.7 traz a ideia geral da abordagem desenvolvida para seguir uma face em vídeos coloridos adquiridos em ambientes não controlados, conforme proposto nesta Tese.

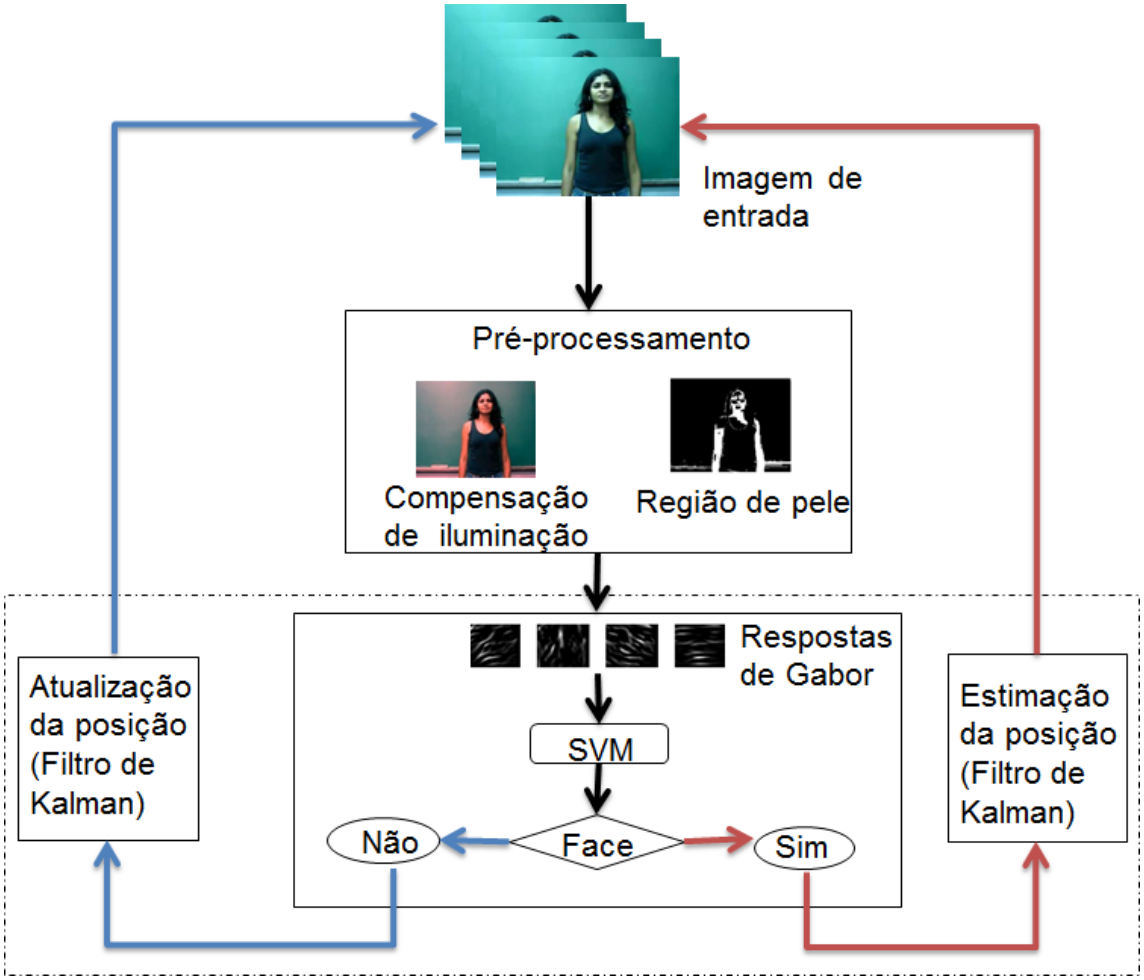


Figura 3.7: Fluxograma do sistema de detecção e seguimento de face proposto.

Capítulo 4

Resultados

Neste capítulo são apresentados os resultados de detecção e seguimento de face em videos reais utilizando o método proposto nesta tese de doutorado. O método apresentado no Capítulo 3 é denominado seguidor dinâmico utilizando vetores de suporte e banco de filtros de Gabor (DSVS). O desempenho do sistema desenvolvido nesta pesquisa foi testado em sequências de vídeo que apresentam iluminação não uniforme, oclusão parcial, rotação, variação de escala. As sequências de vídeos utilizados para testar o desempenho do sistema proposto são vídeos proprietários e alguns obtidos das Honda/UCSD Video Database (Lee, Ho, Yang and Kriegman; 2005) e a sequência David disponibilizado por David Ross (Ross et al.; 2008). Nas seções seguintes os resultados são apresentados e discutidos, considerando-se a metodologia utilizada para alcançá-los.

4.1 Bases para treinamento e teste do SDVS

As imagens de face escolhidas como amostras para treinar a classe face apresentam condições semelhantes, mas não idênticas, às das sequências de vídeo escolhidas para testar o SDVS proposto nesta Tese: variação de iluminação, posição arbitrária da cabeça, rotação, translação, uso de óculos, barba, e diferentes expressões de emoção. A base de face foi escolhida por ser composta de brasileiros, apresentando uma variação de traços étnicos, além de conterem imagens coloridas. Outras bases de faces largamente utilizadas na literatura científica não puderam ser utilizadas por serem imagens não coloridas (Phillips et al.; 1998; Rowley et al.; 1998; Lee, Ho and Kriegman; 2005).

A base de dados de faces FEI (Thomaz and Giraldi; 2010) é um base de dados de faces de brasileiros que contém um conjunto de faces tomadas entre junho de 2005 e

março de 2006 no Laboratório de Inteligência Artificial da FEI (Fundação Educacional Inaciana) em São Bernardo do Campo, São Paulo. Existem 14 imagens para cada um dos 200 indivíduos, num total de 2800 imagens. Todas as imagens são coloridas e tomadas contra um plano de fundo homogêneo branco em posição frontal de frente com rotação de perfil até cerca de 180 graus.

A escala pode variar cerca de 10% e o tamanho original de cada imagem é 640×480. Todas as faces são de estudantes e funcionários da FEI, com idade entre 19 e 40 anos, com aparência, corte de cabelo e adornos distintos. O número de homens e mulheres são exatamente iguais a 100. Também é disponibilizado um subconjunto composto apenas de imagens frontais previamente alinhadas em um modelo comum tal que as informações extraídas daquelas imagens tenham localização aproximada em todos os indivíduos. Neste alinhamento manual, é escolhida aleatoriamente a imagem frontal de um indivíduo como um modelo e a direção dos olhos e nariz escolhidas como uma referência de localização. Todas as imagens frontais deste subconjuto são cortadas para o tamanho 360×260 pixels. Já que o tamanho do subconjunto é igual a 200 e cada indivíduo tem duas imagens frontais (uma com expressão neutra e outra sorridente), há 400 imagens frontais manualmente registradas para avaliação de experimentos em ambiente controlado.

As amostras de face utilizadas para o treinamento do SVM são tomadas do conjunto original de imagens da FEI por apresentar variedade de expressão e posicionamento da face em relação ao plano da câmera. As 2.800 imagens foram recortadas manualmente enquadrando apenas a região de face e redimensionadas para adquirirem a dimensão 30×40. Um exemplo das imagens de face do conjunto original da FEI é exibido na Figura 4.1.

Figura 4.1: Algumas imagens de face da FEI Face Database (Thomaz and Giraldi; 2010) utilizadas para treinar o SVM na etapa de detecção de face do SDVS.

As amostram de padrão não-face foram selecionadas de vídeos não utilizados para teste, gravados no Campus Goiabeiras da UFES e de imagens de textura colhidas na internet. Cada amostra possui a dimensão 30×40 e é colorida.

As sequências de vídeos disponíveis para fins de testes em pesquisas científicas são

destinados a testar seguimento de face para identificação, são capturados em ambientes controlados os rostos estão numa distância em que por características de face não seriam detectadas (*CAVIAR surveillance Dataset*; 2003; Jepson et al.; 2003). Em contrapartida, o problema abordado nesta tese necessita ser testado em vídeos em que os indivíduos se movimentem mais livremente em ambientes sem controle de iluminação ou plano de fundo. Os vídeos capturados no Campus da UFES, com a colaboração de estudantes desta instituição foram capturados para que o SDVS pudesse ser testado adequadamente em ambientes não controlados. Sâo sequências coloridas, internas e externas capturadas em ambientes reais, com uma máquina fotográfica comum, tipo cybershot, mas que apresenta recursos de captura de vídeo. As pessoas nestas sequências de vídeos adquirem posturas e expressões de emoções diferentes, alguns utilizam óculos de grau ou de sol. Os vídeos capturas a uma taxa de 30 fps com quadros de dimensão 240×320. Cada um deles apresenta um desafio para o SDVS, como descrito na Tabela 4.1.

Dos vídeos não proprietários utilizados nos testes, somente a sequência David (Ross et al.; 2008) possui as características desejáveis, já descritas, para testar o SDVS. O homem neste vídeo se movimenta arbitrariamente e muda de ambientes com iluminação não uniforme, esboça perfil total e perfil, usa óculos de grau e os tira durante a sequência. São 770 quadros capturados sob uma taxa de 15fps. Por simplicidade a sequência de vídeo *Handsome Fellow* será chamada de David neste trabalho.

Dois vídeos da Honda/UCSD Video Database (Lee, Ho, Yang and Kriegman; 2005) são utilizados para teste de seguimento do SDVS. Cada sequência de vídeo é capturado em ambiente interno a uma taxa de 15 fps. A resolução de cada vídeo é 640×480. Nos vídeos os indivíduos rotacionam a face e voltam a posição original, sentados em frente a uma câmera, com diferentes velocidades. Os indivíduos se aproximarem e se afastarem da câmera que os filmam, causando grande variação na escala de suas faces. Os vídeos foram capturados com uma câmera SONY DFW-V500 no Computer Vision Laboratory,

Tabela 4.1: Sequências de vídeo proprietários e seus principais desafios para o SDVS.

Video	# de quadros	Desafio
Video1	1.681	Movimento
Video2	1.771	Oclusão parcial
Video3	500	Ambiente externo
Video4	100	Ambiente externo e escala
Video6	301	Tom de pele escuro
Video5	1.025	Perfil total

University of California, San Diego em 2004.

As amostras de face utilizadas para treinar o classificador foram redimensionadas para a dimensão 30×40. Os vídeos de teste também foram redimensionados para que cada quadro adquirisse a dimensão 240×320.

4.2 A detecção da face

Detecção de face é um assunto frequentemente abordado como um problema de duas classes: a dificuldade é a complexidade de definir a classe não-face. Neste trabalho, nós podemos definir o que é face. Isto se faz necessário pelo fato do sistema de seguimento de face apresentado aqui ser aplicado a vídeos coloridos de cenas reais em que o indivíduo nem sempre adquire a postura de frente para a câmera ao ser filmado. Além disto, a face pode estar apenas parcialmente visível em decorrência de oclusão causada por objetos presentes na cena, por sombras produzidas durante a filmagem ou podem sair do campo de visão da câmera. Portanto, neste sistema, considera-se que uma face é a região apresentando pelo menos um olho. A classe não-face é todo o restante do universo capturado nas sequências de vídeo. A etapa de detecção de face precisa apresentar um bom desempenho, pois sem alvo não há o que ser seguido.

Às imagens escolhidas para compor a base de treino para o classificador é aplicado o algoritmo de compensação de iluminação. Após a etapa de pré-processamento das amostras, então, cada amostra da base de treino é passada pelo banco de filtro de Gabor, com 4 parâmetros de orientações e um de frequência, através de uma operação de convolução. A dimensão da resposta do banco de filtro de Gabor é 30×40×4. É obtido o valor máximo absoluto das respostas do banco de filtro de Gabor que possui a mesma dimensão da amostras originais 30×40.

Descrito no capítulo anterior, o classificador utilizado para a etapa de detecção de face do SDVS é um SVM de *kernel* RBF. A dimensão deste vetor de características é 1200, cada máximo absoluto do banco de filtro de Gabor, de dimensão 30 x 40 pixel, foi vetorizado antes de ser apresentado ao SVM. Foram utilizadas cinco mil amostras de face e não-face, sendo 2.500 faces e 2.500 não-faces. A base de dados de rostos de brasileiros disponíveis em FEI Face Database(Thomaz and Giraldi; 2010) foi utilizada para retirar as amostras de face. As amostras de imagens da classe não-face foram retiradas de cenas de videos não utilizados nos testes. Vale ressaltar que na base de faces(Thomaz and Giraldi; 2010) os indivíduos nem sempre estão de frente ou com feições neutras.

A detecção de face tem duas medidas para avaliação (Fawcett; 2006): taxa de falso positivo (TFP) e taxa de verdadeiro positivo (TVP). Um falso positivo significa que uma amostra da classe não-face é mal classificada como pertencente à classe face. Um verdadeiro positivo significa que uma amostra de face é corretamente classificada como face. Para mensurar estas duas medidas simultaneamente uma curva ROC (do inglês *Receiver Operating Characteristic*) é utilizada nesta Tese. As curvas ROC são uma ferramenta útil e para visualização e avaliação quantitativa de classificadores binários. Desta forma, o desempenho do classificador se torna uma curva no plano TFP-TVP. As curvas ROC são capazes de prover uma medida mais rica do desempenho de classificadores do que medidas como precisão ou taxa de erro (Fawcett; 2006). O eixo horizontal exibe a TFP e o eixo vertical a TVP.

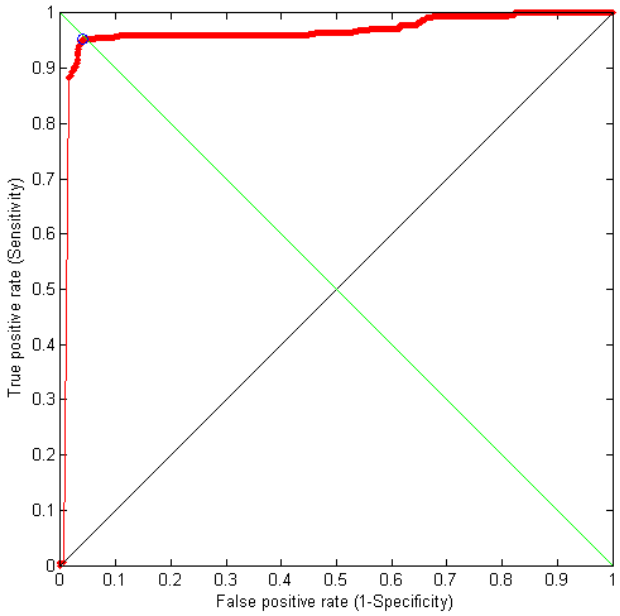


Figura 4.2: Curva ROC para o detector de faces utilizando SVM com *kernel* RBF.

A curva ROC evidencia uma taxa de acerto de 95 por cento obtido ao testar a detecção de face em 400 amostras de face tiradas dos vídeos Video2, David, Honda2 e Alex e 600 não-faces tiradas de imagens de textura conseguidas da internet. A taxa de 95% de amostras de face corretamente classificadas é considerada satisfatória no presente trabalho. É possível observar que a área sob a curva ROC, AUC (do inglês *Area Under the Curve*), representa uma medida de avaliação do desempenho do classificador que confirma um bom desempenho. Quanto mais à esquerda se encontra a curva ROC, consequentemente quanto menor a área acima da curva ROC, maior é a AUC. A forma adquirida pelo gráfico implica em uma elevada taxa de verdadeiros positivos e baixa taxa de falsos positivos. Portanto, o classificador está classificando corretamente as classes face e não-face.

A tabela de contingência ou matriz de confusão (Fawcett; 2006) correspondente à curva ROC da Figura 4.2, exibida na Tabela 4.2, revela quantas amostras de face e não-face foram corretamente classificadas e quantas não o foram. Os valores na diagonal principal representam as classificações corretas e a diagonal secundária os erros ou confusões entre faces e não-faces. Das 400 faces, 381 foram corretamente classificadas como face (FV), enquanto que, das 600 amostras de não-face, 479 foram corretamente classificadas como não sendo face (NFV). Então, 19 faces foram erroneamente classificadas como não-face (NFF) e 21 amostras de textura foram classificadas como face (FF).

Tabela 4.2: Matriz de confusaõ para o teste do classificador SVM RBF.

	# Face	# Não-Face	
# Faces classificadas	FV 381	FF 21	Valor Preditivo Positivo FV/(FV+FF) 94,7%
# Não-faces classificadas	NFF 19	NFV 479	Valor Preditivo Negativo NFV/(NFF+NFV) 96%
	sensibilidade FV/(FV+NFF) 95%	especificidade NFV/(FF+NFV) 95,8%	

A partir da matriz de confusão é possível calcular algumas medidas, como mostrado na própria Tabela 4.2. A sensibilidade do classificador é expressa em porcentagem, dividindo-se o número de faces corretamente classificadas pelo número de faces. Também chamada de *recall* ou taxa de verdadeiro positivo, ela mede a capacidade do classificador de indicar a presença da característica entre os elementos que a possuem. Por outro lado, a especificidade do classificador é medida, em porcentagem, dividindo-se o número de não-faces corretamente classificadas pelo número de não-faces. Ela mede a capacidade do sistema de identificar a ausência da característica entre os elementos que realmente não a possuem. O Valor Preditivo Positivo, também conhecido como *precision*, indica a proporção dos elementos classificados como possuidores da característica que realmente a possuem. Já o Valor Preditivo Negativo indica a proporção dos verdadeiros negativos em relação a todas as classificações negativas.

O desempenho satisfatório da detecção de face é atribuída, em primeira instância, à escolha do vetor de características discriminativo. As respostas de Gabor realçam regiões da imagem de face de acordo com os parâmetros de frequência e orientação utilizados. Estas respostas de Gabor terão valor alto apenas em locais específicos, que realçam a pose e escala adquirida pela face na imagem original de amostra de face. Assim, é possível utilizar, em vez das quatro respostas do banco de filtro de Gabor,

apenas o valor máximo absoluto obtido de cada filtro de Gabor.Pontanto, o máximo absoluto das resposta do banco de filtros de Gabor se adequada bem ao problema tratado nesta tese. A escolha de extrair características de Gabor da aparência global da face favorece a detecção correta de faces mesmo que estas estejam em oclusão parcial. Mesmo que parte da face esteja sob oclusão, ainda é possível detectar que aquele pedaço da imagem é uma face. O desempenho da detecção de face está também na associação do vetor de características discriminativo com um classificador SVM de *kernel* RBF, considerado o mais adequando classificador binário na literatura.

4.3 *Framework* Viola & Jones

O SDVS é comparado com o *framework* proposto por Viola and Jones (2001). Esta escolha é quase mandatória pelo fato do *framework* (Viola and Jones; 2001) ser largamente utilizado na detecção e rastreio de face em vídeo, apesar de haver propostas mais recentes na literatura (He et al.; 2002; Ross et al.; 2008; Babenko et al.; 2011; Zhang et al.; 2012). O algoritmo é utilizando para fins de comparação com a abordagem proposta nesta tese. Sua descrição é apresentada a seguir e está disponível na biblioteca de funções de visão computacional OpenCV (Bradski; 2000). O código foi executado a partir do $MATLAB^{®}$ por meio da geração de um arquivo que permite a chamada de códigos implementados na linguagem C^{++}, arquivos *mex*.

O algoritmo proposto por Viola and Jones (2001) varre a imagem a fim de detectar face. Originalmente, este algoritmo detecta 15 faces por segundo. Por prover uma solução de detecção em tempo real, ele é largamente utilizado em sequências de vídeo para localizar face. A solução proposta por Viola and Jones (2001) calcula a imagem integral de uma imagem de entrada em nível de cinza. Nesta técnica cada píxel $P(x, y)$ é substituido pelo somatório dos valores de todos os píxels localizados acima e à esquerda de $P(x, y)$. Desta forma é possível calcular o valor da integral de uma região D delimitada pelos pixels $(P1, P2, P3, P4)$ da imagem integral rapidamente pela relação $P4 - P3 - P2 + P1$.

A detecção do objeto é realizada através da análise de características de regiões retangulares na imagem, havendo similaridades com as funções base de Haar utilizada na Transformada *Wavelet* de *Haar* (Gonzalez and Woods; 2006). As caracteríscias , apresentadas na Figura 4.3, são as que podem ser detectadas: (a) característica de dois retângulos: diferença entre a soma de pixels entre duas regiões retangulares; (b) característica de três retângulos: diferença entre a soma de pixels de duas regiões retangulares laterais e uma região central; (c) característica de quatro retângulos: diferença

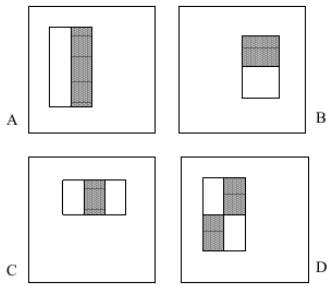


Figura 4.3: Exemplos de características que podem ser detectadas no *framework* (Viola and Jones; 2001): A e B com dois retângulos, C com três retângulos e D com quatro retângulos.

entre a soma de pixels entre duas regiões retangulares de uma diagonal e a soma de pixels das duas regiões retangulares na diagonal oposta.

O processo de treinamento do classificador que determina quais regiões são alvos em potencial, neste caso as regiões de faces, é realizado através de estímulos com exemplos positivos e exemplos negativos. Apesar de existir uma variedade de possibilidades de características, somente poucas são suficientes para determinar a face. Viola and Jones (2001) utilizam uma variação do AdaBoost (Freund and Schapire; 1995) tanto para selecionar as características relevantes quanto para treinar o classificador. O algoritmo Adaboost é utilizado para melhorar o desempenho de um classificador através da combinação de funções de classificação fracas para formar um classificador mais forte. Por exemplo, após o processo de treinamento de um classificador fraco ser encerrado, é realizado uma atribuição de pesos aos exemplos na entrada de um novo classificador com intenção de enfatizar os exemplos que foram incorretamente classificados a princípio.

A convergência dos classificadores fracos para o classificador forte é utilizada para obter o menor número de erro de classificação. A cada iteração do Adaboost são atribuídos pesos maiores para algumas características que são mais representativas para os exemplos de entrada (faces). Como consequência, obtém-se um classificador com poucas características e capaz de detectar o objeto de interesse. O erro decai de forma exponencial a cada iteração e utilizando uma margem maior de exemplos ocorre uma maior generalização. Na prática, é escolhida a região retangular que melhor separa os exemplos negativos dos positivos (escolha das características que serão usadas no classificador de face).

O classificador final do Viola-Jones é obtido como a combinação linear de alguns classificadores associados a diferentes características (regiões retangulares) e o peso de cada classificador é inversamente proporcional à taxa de erro obtida. Observa-se que há

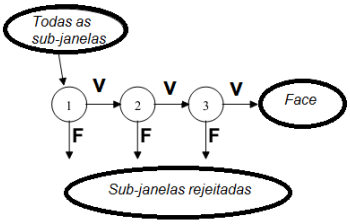


Figura 4.4: Representação esquemática de uma cascata de detectores como o proposto por Viola and Jones (2001). V são as subjanelas da imagem detectadas como face e N são as sub-janelas da imagem detectadas como não-face.

um interesse em se trabalhar com valores de intensidade/luminosidade do pixel/área.

Para a detecção de faces, a saída do algoritmo de aprendizagem de seleção de características escolhe regiões centradas nos olhos. A primeira é a região que compreende olhos e a parte abaixo dos olhos. Isso se deve ao fato de que regiões dos olhos são normalmente mais escuras que a região da bochecha. A segunda característica envolve os olhos e a ponta do nariz, pois os olhos são mais escuros que a ponta do nariz, em geral. Portanto, é coerente afirmar que o algoritmo de Viola-Jones para o caso de detecção de faces procura, a princípio, os olhos.

Nos testes realizados por Viola-Jones foi alcançado uma taxa de acerto de 95% para o banco de testes . A varredura da imagem começa com sub-janelas de 24 $\times$ 24 pixels e percorre-se toda a imagem. Em seguida, as janelas são aumentadas por um fator de 1,25 e o procedimento se repete até o tamanho da janela atingir o tamanho total da imagem. Observa-se que se faz necessário um procedimento de converter várias entradas de face localizadas para a mesma face em diferentes resoluções de janela para apenas uma face e uma localização.

É utilizada uma estrutura de árvore de decisão onde há vários classificadores em cascata (*Attentional Cascade*) para reduzir o tempo de processamento. A chave da árvore de decisão é construir um classificador capaz de maximizar a rejeição das imagens negativas (que não contém face) e simultaneamente capaz de detectar a maioria das imagens positivas (com face). A face só é detectada se todos os estágios em cascata derem positivos, como no esquema da Figura 4.4. O algoritmo para ao primeiro caso negativo.

A base de vídeos usada para treino e teste é a MIT+CMU (Rowley et al.; 1998) com 130 imagens e 507 faces e o *framework* processa 15 quadros por segundo.

4.4 Seguimento da face

O resultado do método Seguidor Dinâmico com Vetores de Suporte é apresentado a seguir. O localização da face é umas das etapas principais em sistemas de vigilância. O problema é seguir e definir a localização da face de uma pessoa em imagens coloridas. As sequências de vídeo proprietárias e os vídeo Honda/UCSD Video Database (Lee, Ho, Yang and Kriegman; 2005) e o vídeo de Ross et al. (2008) utilizados para testar o SDVS são apresentadas a seguir. Nos experimentos a busca exaustiva somente nas regiões de cor de pele é realizada no primeiro quadro, já que a posição da face alvo é desconhecida. Nenhuma face foi detectada em *frames* anteriores. A partir do segundo quadro, o método proposto é aplicado para estimar, com o Filtro de Kalman, o ponto de localização da face. Contudo, se nenhuma observação se torna acessível naquela localização o filtro de Kalman atualiza a posição da face para o próximo quadro e a busca pela face é aplicado na vizinhança da posição do alvo no próximo quadro, limitado a uma janela de 80×60 pixels. O que a literatura apresenta são soluções de seguimento que esperam novamente o surgimento da face, votlando ao passo inicial. A região de vizinhança é somente esta janela de busca no conjunto de pixels de cor da pele. Retângulos vermelhos mostram a região seguida, dando a estimação do filtro de Kalman. Neste documento são mostrados resultados da detecção e seguimento correspondentes às sequências de vídeos escolhidas por apresentarem movimentos mais complexos, variação de escala, oclusão parcial e mudança na visualização da face para avaliação do SDVS como descrito na Tabela 4.3. Para demonstrar o desempenho do seguimento de face do SDVS será avaliada a trajetória realizada pelo sistema proposto e a robustez do sistema em relação as oclusões de face encontrados nas sequências de vídeo da Tabela 4.3 São exibidos instantes do seguimento da face nos vídeo e a robustez do sistema em relação à recuperação da localização de face em oclusão é discutida. Este fato pode ser confirmado pelos dados exibidos na Tabela 4.4. A abordagem proposta nesta tese é comparada com o método de detecção de face proposto por Viola and Jones (2001). Na sequência, é verificada a efetividade do método proposto sob rotação e oclusão parcial de face. A sequência de vídeo usada no teste foi a de nome Video1, para a qual, seis instantes são mostrados na Figura 4.5, com a face detectada mostrada com um retângulo vermelho. Do quadro dois ao quadro 60 o homem se move na frente da câmera. Ele se move de um lado para outro do quadro. Após alguns quadros ele também move o corpo na direção vertical (do quadro 200 ao 300). Observa-se também que ele modifica a visualização de sua face: no quadro 200 o homem esboça um perfil parcial, e então ele se move novamente até uma visão frontal no quadro 300. Iniciando no quadro de número 1.000 o homem, parado, move a face até alcançar perfil completo, no quadro 1.300. Mesmo com todas estas alterações a face do homem é corretamente

Tabela 4.3: Descrição dos principais desafios apresentados nos vídeos proprietários (ver Tabela 4.1), nos vídeos Honda/UCSD Video Database (Lee, Ho, Yang and Kriegman; 2005) e no vídeo David (Ross et al.; 2008), utilizados para testar o SDVS.

Video	Quadros	Desafio
Video1 (proprietário)	1.681	Movimento
Video2 (proprietário)	1.771	Oclusão parcial
Video3 (proprietário)	500	Ambiente externo
Video4 (proprietário)	100	Ambiente externo e escala
Video6 (proprietário)	301	Tom de pele escuro
Video5 (proprietário)	1.025	Perfil total
Honda1 (Lee, Ho, Yang and Kriegman; 2005)	500	Iluminação
Honda2 (Lee, Ho, Yang and Kriegman; 2005)	500	Iluminação
David (Ross et al.; 2008)	770	Iluminação e movimento

detectada e seguida neste vídeo.

O próximo teste foi feito utilizando a sequência de vídeo Video2 (quatro instantes são mostrados na Figura 4.6, em que os retângulos vermelhos marcam as faces detectadas, uma vez mais). Nesta sequência é verificado o desempenho do sistema proposto sob oclusão parcial da face. O método detecta corretamente a face do quadro 1 (visão frontal da face) até o quadro 261 (face inclinada), como mostrado nos quatro instantes na Figura 4.6. A face alvo é corretamente seguida através dos quadros, até mesmo sob oclusão parcial.

Deve ser ressaltado que, a respeito de todas as restrições presentes nas sequências de vídeo utilizadas devido aos movimentos feitos pelos indivíduos, o método de seguimento aqui proposto foi capaz de seguir efetivamente a face de uma pessoa. O terceiro teste (sequência de vídeo Video3) foi realizado seguindo uma face em um ambiente externo, como mostrado na Figura 4.8. A robustez às condições do mundo real é apresentada. O homem na sequência de vídeo se movimenta para longe da câmera. O plano de fundo da cena apresenta diferentes texturas e a iluminação está frequentemente mudando devido às sombras ao redor do sujeito. Os instantes desta sequência de video apresentaram variação de escala e a câmera não estava fixa no momento da captura.

A sequência Honda2 escolhida para os testes é o vídeo Honda2 da base Honda/UCSD Video Database (Lee, Ho, Yang and Kriegman; 2005). Uma mulher, nos instantes mostrados na Figura 4.7, está sentada em frente a uma câmera em um escritório. O plano de fundo aqui é complexo, e é possível notar algumas janelas na sala. Isto significa que o ambiente recebe iluminação natural e artificial ao mesmo tempo. Neste teste o

Figura 4.5: Seis instantes (quadros 2, 60, 200, 300, 1.000 e 1.300, em zigzag) para a sequência de vídeo Video1 (Tabela 4.3), um caso mostrando rotação da face e deslocamento na vertical.

método proposto detecta e segue a face através de mudança de brilho. Finalmente, o desempenho de detecção e seguimento é bem sucedido mesmo quando a pessoa olha para alguns pontos na parede ou olha para frente e para cima. O sequência de Vídeo Honda1, na Figura 4.9, é outra sequência de vídeo da Honda/UCSD Video Database. O indivíduo está sentado em frente a uma câmera e apresenta movimentos semelhante ao vídeo Honda2. O SDVS consegue seguir satisfatoriamente o homem em Honda1. A velocidade do seguidor é 3 quadros por segundo em um PC Dual Core. Esta taxa inclui todo o processo: ler a imagem, detectar pele e face, estimar a próxima posição, mostrar os resultados no quadro, exibir retângulos para a face detectada e a posição para o próximo quadro respectivamente, com todo o sistema desenvolvido rodando na plataforma $MATLAB^{®}$.

Uma avaliação considerando a trajetória real do alvo no vídeo e a posição estimada pelo método proposto é apresentada a seguir. Primeiro, a posição do nariz de cada indivíduo é considerada como a posição real da face. Este ponto é assumido pois é também utilizado no estágio de corte das faces para treino. Estes pontos foram marcados quadro por quadro manualmente. Após uma observação se tornar acessível, o filtro de Kalman estima a posição do alvo para o próximo quadro e uma nova observação é obtida na vizinhança de cor da pele daquele ponto. Na Figura 4.5 é possível observar os

Figura 4.6: Sequência de vídeo de teste Video2 (quadros 69, 200, 261 e 298, acompanhar em zig-zag). A pessoa movimenta o rosto em frente à câmera e o esconde parcialmente com um papel.

Figura 4.7: Sequência de vídeo Honda2, adquirida do banco de vídeos HONDA, indivíduo sentado em frente à câmera, movendo a cabeça de um lado para o outro, para cima e para baixo.

resultados de seguimento para as sequências de vídeo obtidas neste trabalho. Deve ser observado que as posições estimadas no gráfico sofreram um deslocamento, em relação à posição real, o qual não é um inconveniente para o método proposto neste trabalho. O deslocamento é esperado e se deve ao fato de que o filtro de Kalman seguir uma coordenada e possuir um passo de correção da obserção visto na Seção 2.5. De acordo

Figura 4.8: Quatro instantes do seguidor de face para a terceira sequência de vídeo da Tabela 4.3, que mostra a detecção e seguimento da face. O indivíduo se aproxima e se afasta da câmera em ambiente externo, no Campus de Goiabeiras da UFES.

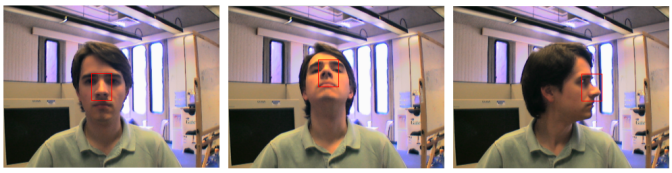


Figura 4.9: Três instantes do seguidor de face na sequência Honda1.

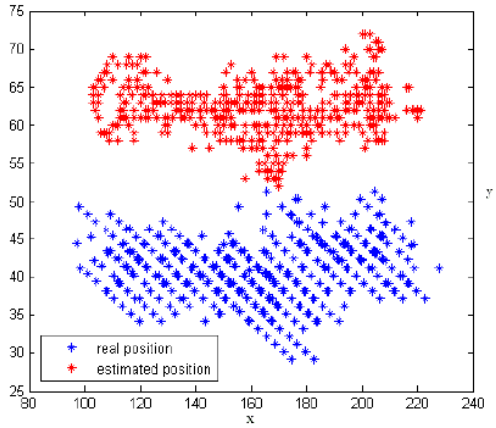


Figura 4.10: Gráfico da localização x,y das trajetórias real (azul) e estimada (vermelho) para a sequência de vídeo Video1.

com o trabalho apresentado, as faces são encontradas em uma janela centralizada na posição produzida pelo filtro de Kalman e as faces são corretamente detectadas pelo (SDVS) apresentado neste proposta de tese. Resultados similares foram observados

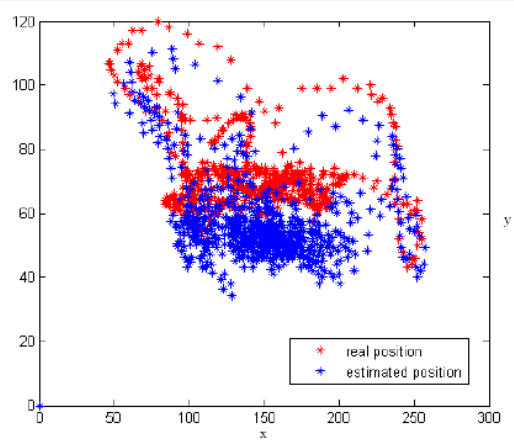


Figura 4.11: Gráfico da localização x,y das trajetórias real (azul) e estimada (vermelho) para a sequência de vídeo Honda2.

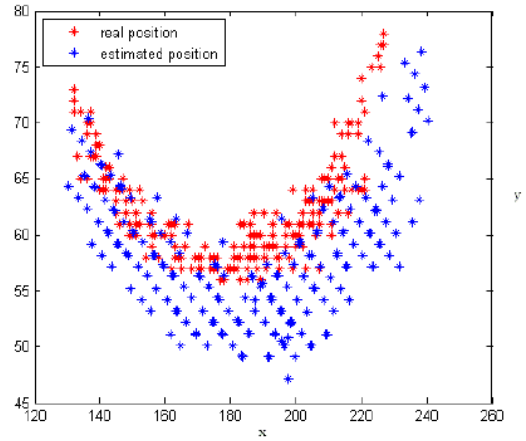


Figura 4.12: Gráfico da localização x,y das trajetórias real (azul) e estimada (vermelho) para a sequência de vídeo Video3.

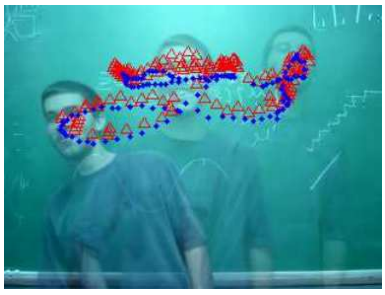


Figura 4.13: Sobreposição das trajetórias real (azul) e estimada (vermelho) para a sequência de vídeo Video1.

as demais sequências de vídeo da Tabela 4.3. Os gráficos para estas trajetórias são exibidas nas Figuras 4.10,4.11 e 4.12. A sobreposição das trajetórias sobre as imagens

Figura 4.14: Sobreposição das trajetórias real (azul) e estimada (vermelho) para a sequência de vídeo Honda2.

Figura 4.15: Sobreposição das trajetórias real (azul) e estimada (vermelho) para a sequência de vídeo número Video3.

dos respectivos vídeos pode se observada, respectivamente nas Figuras 4.13,4.14 e 4.15.

A moça na sequência Video5, da qual alguns quadros podem ser observados na Figura 4.16, se move mais lentamente e permanece de perfil total. Algumas vezes o rosto é completamente escondido pelo seu cabelo. O filtro de Kalman, atualizou a localização da face, nos momentos em que o alvo não foi detectado pelo SVM.

Os vídeos Video4, exibindo na Figura 4.17, Video6, na Figura 4.18 e Davi, na Figura 4.19, são sequências que representam desafios maiores para o sistema proposto nesta tese. A sequência Jorge Figura é um vídeo gravando em ambiente externo em que o indivíduo se move bem como a câmera, para acompanhá-lo. A face neste vídeo adquire dimensões pequenas e em seguidas, dimensão que chega a ocupar quase completamente a área de captura da câmera. Nestes instantes o SVM detecta um falso positivo, o filtro de Kalman recebe a observação e estima a posição da face para o próximo quadro da sequência. A busca por uma face é realizada na janela de busca centrada na posição estimada pelo Kalman. É possível que o SDVS, nesta situação esteja seguindo um falso positivo por alguns quadros seguidos. Porém, o uso de um SVM bem treinado evita que esta situação se prolongue. À janela de busca, a cada quadro, é aplicado o SVM para confirma se é realmente face o pedaço da imagem sendo investigado, cujo centro

Rowena.png

Figura 4.16: Seis instantes de seguimento de face para a sequência Video5 (Tabela 4.3), um caso com pequena área de face, vista lateral.

Figura 4.17: Seis instantes do seguidor de face para a quarta sequência de vídeo da Tabela 4.3, que mostra a detecção e seguimento da face.

é a posição de localização estimada pelo filtro de Kalman.

Na sequência de vídeo Davi, o sujeito passa por vários ambientes. Ele sai de um corredor completamente escuro e se encaminha para uma sala maior iluminada. Os primeiros 220 quadros da sequência são capturados no corredor escuro. David usa óculos de grau e os tira por alguns instantes. Mesmo na sala iluminada é possível perceber a formação de sobras sobre o indivíduo. No início do vídeo a câmera está mais próxima da face e toma distância ao longo da sequência. Nos instantes em que a

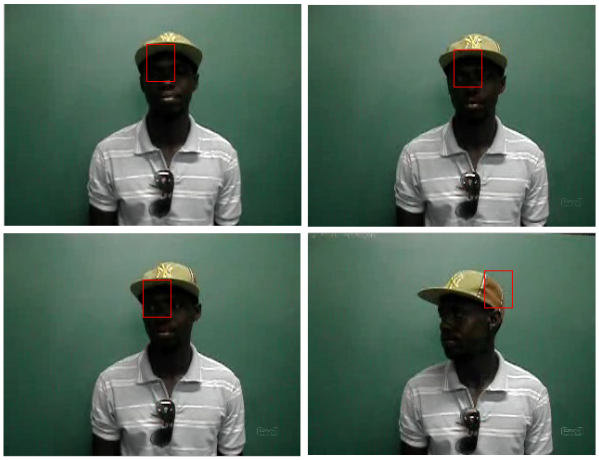


Figura 4.18: Quatro instantes para a sequência Video6 (veja Tabela 4.3),um caso de cor da pele extremamente escura.

face é aproximadamente 60×80, a região de face detectada pelo SVM é o olho. Ainda que, neste vídeo, o tamanho da face seja o dobro, a disposição do filtro de Gabor, de acordo com a frequência escolhida, permite que parte da face seja detectada. Este comportamento pode ser observado nos vídeos com escala da face ou oclusão parcial.

O vídeo Video6 é uma sequência de imagens em que o indivíduo filmado é um homem com tom de pele muito escuro. Nesta sequência de imagens o seguimento da face foi comprometido por regiões de tons de pele que não foram classificadas como tal, gerando regiões de falso negativos para pele. Mesmo propondo, nesta tese, uma faixa de cor para tons de pele mais escuros, estes de pele são menos detectados que os tons pardos. Para que seja possível realizar busca por face em tons de pele mais escuros não foi utilizado limite para a tamanho mínimo da região de pele a ser varrida pelo SDVS. Desta forma, pequenas regiões de pele, em que é possível haver uma face, são também exploradas.

A Tabela 4.4 exibe o número de *frames* de cada sequência de vídeo, o número de quadros em que as faces não foram detectadas pelo SDVS e pelo Viola and Jones (2001) e o número de quadros em que as faces foram recuperadas pelo SDVS. Viola e Jones não trata a dinâmica do seguimento de face, ele detecta ou não face a cada *frame*, varrendo a imagem em diversas escalas. Por este motivo não é possível calcular o número de faces recuperadas. Apesar de ser rápido, o *framework* de Viola and Jones (2001) apresenta muitos falsos negativos. Isto se deve ao fato da limitação de as características retangulares apresentam somente serem adequadas para detectar face com variação de ± 10 graus da posição vertical. Outra possibilidade de falha pode ser devido às bruscas mudanças de iluminação, já que a técnica de imagem retangular

Figura 4.19: Oito instantes para a sequência David (veja Tabela 4.3), mudança de iluminação, escala, movimento da câmera.

buca por diferença de nível de cinza para identificar a face dentro de cada imagem aoresentada.

A *wavelet* Haar apresenta a característica de ser variante ao deslocamento (Mallat; 2008), o que não é desejável para detecção de face em pose arbitrária. Sob este aspecto a magnitude das respostas de Gabor levam vantagem por serem invariates a translação, rotação e escala. Diferente da possibilidade das respostas de filtro de Gabor, as características retangulares apresentam somente orientação vertical, horizontal e diagonal.

Tabela 4.4: Robustez do SDVS, medida pelo número de faces recuperadas no quadro seguinte.

Video	Número de quadros	Faces Perdidas	Faces Perdidas (Viola and Jones (2001))	Faces recuperadas no quadro seguinte SDVS	%
Video1	1.681	70	90	**69**	**99**%
Video2	1.771	63	338	**62**	**98**%
Video3	500	110	145	**90**	**81**%
Video4	100	20	53	**18**	**90**%
Video6	301	50	246	**40**	**80**%
Video5	1.025	70	663	**68**	**97**%
Honda1	500	100	220	**95**	**95**%
Honda2	500	100	201	**95**	**95**%
David	770	100	201	**90**	**90**%

A face pode não ser detectada por algum tipo oclusão ou pela ocorrência de um falso negativo. Nestes caso o filtro de Kalman não tem uma observação acessível. A associação da etapa de detecção com o filtro de Kalman evita que o SDVS perca a localização da face no próximo quadro da sequência de vídeo sendo testada. Isto é possível pois o filtro de Kalman pode atualizar a posição para o próximo *frame*, como descrito no Capítulo 2. Em algumas sequências de vídeo testadas a porcentagem de faces não detectadas recuperadas pelo SDVS passa dos 90%. É o caso dos vídeos Video1, Video2, Video4, Video5, Honda1 e Honda2. Enquanto outros vídeos tem 80%(Video6) e 81%(Video3) de faces perdidas recuperadas pelo SDVS.

Capítulo 5

Conclusões

Nesta Tese de Doutorado foi apresentado o Seguidor Dinâmico com Vetores de Suporte - SDVS e filtros de Gabor, proposto para resolver o problema de seguimento de face em vídeos coloridos adquiridos em ambientes não controlados. O problema em si foi apresentado no Capítulo 1, onde os métodos propostos para resolver detecção e seguimento de face foram apresentados como uma revisão bibliográfica. Para resolver o problema apresentado foram escolhidas algumas técnicas matemáticas, utilizadas em cada etapa do SDVS. No Capítulo 2 a teoria matemática que envolve as técnicas eleitas é apresentada. O SDVS completo, com a descrição de como suas etapas cooperam entre si para seguimento de face é apresentado no Capítulo 3. No capítulo 4 são apresentados resultados alcançados aplicando-se o sistema completo (inclusive com o SVM RBF treinado para reconhecer uma face humana) no processamento de vídeos capturados no campus da Universidade Federal do Espírito Santo e de vídeos obtidos de bases de dados disponíveis gratuitamente na web para testes. Na sequência, neste capítulo são feitas as considerações finais a respeito dos resultados alcançados para seguimento de face utilizando o sistema SDVS aqui proposto, assim como são sugeridas perspectivas para trabalhos futuros.

5.1 Considerações Finais

A respeito da abordagem para detecção e seguimento de face proposta (SDVS), pode-se pontuar os seguintes itens para conclusão desta Tese:

- a observação da face é obtida pela detecção da mesma com um SVM RBF apenas nas regiões de pele, diminuindo a carga computacional, por evitar a busca em toda a imagem;

- o sistema proposto estima a posição da face no próximo quadro, a partir da observação obtida no quadro atual, utilizando um filtro de Kalman discreto e um modelo linear de movimento (velocidade constante);

- a predição feita com o filtro de Kalman é utilizada como ponto de partida para a busca da face no próximo quadro, evitando assim que a detecção da face seja feita pela varredura em toda a região cor de pele no quadro de imagem, reduzindo ainda mais o custo computacional. O filtro de Kalman também é utilizado para atualizar a localização da posição da face no próximo quadro, na ausência de observação;

- o sistema proposto não utiliza detecção de características como olhos e boca. Note-se que em casos em que os indivíduos usam óculos ou em que haja algo que obstrua estas características, o detector baseado em características falha. Assim, onde os detectores baseados em características falham o SDVS pode não falhar (em geral falha menos, como mostram os experimentos aqui discutidos);

- o sistema proposto não utiliza as proporções do rosto, que restringem o sistema a imagens frontais. Assim, o SDVS pode detectar a face mesmo em imagens em diversas orientações;

- o sistema proposto não utiliza modelo estatístico baseado em cor, o que poderia confundir o alvo de fundo com objetos da mesma cor da pela da face;

- a compensação de iluminação utilizada no SDVS é um passo que determina o desempenho de qualquer sistema para o problema proposto; isso é evidente nos vídeos utilizados para teste, que são de baixa qualidade, produzidos por câmera não profissional;

- o cenário dos vídeos testados apresenta condições comuns a um ambiente não controlado;

- os indivíduos nos vídeos mudam as expressões faciais naturalmente;

- a compensação de iluminação utilizada neste trabalho utiliza apenas o espaço de cor RGB, com os valores não normalizados. Testes feitos com o modelo de cor HSV não apresentaram resultados satisfatórios, quer nos vídeos correspondentes a ambientes internos quer naqueles correspondentes a ambientes externos. Porém, houve detecção de pele em alguns vídeos externos, mas dependente das regiões com iluminação solar direta.

Em síntese, foi concebido, implementado e testado um sistema completo de detecção e seguimento de face em vídeos coloridos capturados em ambientes não controlados, o qual se mostra robusto à oclusão parcial e à iluminação não controlada. Entretanto, o sistema apresenta alguma vulnerabilidade devido à dependência, inerente a sistemas de seguimento em ambientes não controlados, da etapa seguinte em relação à etapa anterior. Assim, a abordagem proposta falha se não houver região de pele visível (portanto, se não for detectada uma face), quando há o seguimento de um falso positivo, ou quando há oclusão parcial por tempo prolongado. Ele também falha quando a face adquire dimensão além dos limites selecionados na sua concepção (os tamanhos de janela, especificamente). Nesse caso, o sistema tem capacidade de recuperar-se se a face voltar a uma dimensão dentro dos parâmetros estabelecidos. De qualquer forma, os parâmetros podem ser modificados, o que corresponderia a uma recalibração do sistema, para trabalhar com faces de dimensão maior (escala). Por outro lado, o sistema tem o aspecto positivo de abordar o problema de iluminação não controlada, bem como a dinâmica de uma sequência de vídeo, apresentando considerável robustez no seguimento de face em pose arbitrária, com iluminação não uniforme, e com presença de oclusão, chegando a apresentar 99% de acerto no seguimento da face, considerando o total de quadros da sequência de vídeo.

5.2 Trabalhos Futuros

Decorrente do trabalho de pesquisa realizado durante este doutoramento, sugere-se, como futuros trabalhos, os seguintes temas:

- investigar outras técnicas que permitam detectar os diversos tons de pele humana;
- investigar técnicas que possam ser utilizadas para identificar os indivíduos seguidos nas sequências de vídeo adquiridas em ambientes não controlados;
- investigar outras soluções para a seleção de características, para gerar um vetor de características representativo mas de dimensões menores;
- investigar técnicas que possam contornar o seguimento de falsos positivos no seguimento de face em sequências de vídeo correspondente a ambientes não controlados.
- investigar estratégias para solucionar o problema de escala da face.

Referências Bibliográficas

Agarwal, V., Abidi, B. R., Koschan, A. and Abidi, M. A. (2006). An overview of color constancy algorithms, *Journal of Pattern Recognition Research* **1**(1): 42–54.

Arulampalam, M. S., Maskell, S., Gordon, N. and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, *IEEE TRANSACTIONS ON SIGNAL PROCESSING* **50**: 174–188.

Babenko, B., Yang, M.-H. and Belongie, S. (2011). Robust object tracking with online multiple instance learning, *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(8): 1619–1632.

Belhumeur, P. N., Hespanha, J. a. P. and Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(7): 711–720.

Birchfield, S. (1997). An elliptical head tracker, *Proceedings of the 31st Asilomar Conference on Signals, Systems, and Computers*, pp. 1710–1714.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Bradski, G. (2000). The OpenCV Library, *Dr. Dobb's Journal of Software Tools* .

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition, *Data Min. Knowl. Discov.* **2**(2): 121–167.

Castaneda, B., Luzanov, Y. and Cockburn, J. C. (2004). Implementation of a modular real-time feature-based architecture applied to visual face tracking, *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, Vol. 04, IEEE Computer Society, Washington, DC, USA, pp. 167–170.

CAVIAR surveillance Dataset (2003). EC Funded CAVIAR project/IST 2001 37540. **URL:** *http://homepages.inf.ed.ac.uk/rbf/CAVIAR/*

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines, *ACM Trans. Intell. Syst. Technol.* **2**(3): 27:1–27:27.

Chellappa, R. and Zhou, S. (2002). Bayesian methods for face recognition from video, *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, IEEE, pp. 4068–4071.

Chen, L. and Grecos, C. (2005). A fast skin region detector for colour images, *IEE International Conference on Visual Information Engineering (VIE 2005)*, Vol. 2005, IEE Conf. Pub., pp. 195–201.

Chen, Z. (2003). Bayesian filtering: From kalman filters to particle filters, and beyond, *Technical report*, Adaptive Systems Lab., McMaster University, Hamilton, ON, Canadá.

Cipra, T. and Romera, R. (1997). Kalman filter with outliers and missing observations, *TEST: An Official Journal of the Spanish Society of Statistics and Operations Research* **6**(2): 379–395.

Comaniciu, D. and Meer, P. (1997). Robust analysis of feature spaces: color image segmentation, *Proc. IEEE Computer Society Conf Computer Vision and Pattern Recognition*, pp. 750–755.

Dadgostar, F., Sarrafzadeh, A. and Overmyer, S. P. (2005). Face tracking using mean-shift algorithm: a fuzzy approach for boundary detection, *Proceedings of the First international conference on Affective Computing and Intelligent Interaction*, ACII'05, Springer-Verlag, Berlin, Heidelberg, pp. 56–63.

Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters, *J. Opt. Soc. Am. A* **2**(7): 1160–1169.

de Almeida, O. C. P. (2006). *Técnicas de processamento de imagens para localização e reconhecimento de faces*, Master's thesis, ICMC-Universidade de São Paulo.

de Campos, T. E. (2001). *Técnicas de Seleção de Características com Aplicações em Reconhecimento de Faces*, PhD thesis, IME-Universidade de São Paulo.

de Campos, T. E., Feris, R. S. and Cesar-Junior, R. M. (2000). A framework for face recognition from video sequences using gwn and eigenfeature selection, *Proceedings of WAICV'2000 Workshop on Artificial Intelligence and Computer Vision*, pp. 141–145.

Do, H.-C., You, J.-Y. and Chien, S.-I. (2007). Skin color detection through estimation and conversion of illuminant color under various illuminations, *IEEE Trans. on Consum. Electron.* **53**(3): 1103–1108.

Ebner, M. (2007). *Color Constancy*, The Wiley-IS&T Series in Imaging Science and Technology, Wiley.

Faux, F. and Luthon, F. (2012). Theory of evidence for face detection and tracking, *Int. J. Approx. Reasoning* **53**(5): 728–746.

Fawcett, T. (2006). An introduction to ROC analysis, *Pattern Recogn. Lett.* **27**(8): 861–874.

Foytik, J., Sankaran, P. and Asari, V. K. (2011). Tracking and recognizing multiple faces using kalman filter and modularpca, *Procedia CS* **6**: 256–261.

Freund, Y. and Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting, *Proceedings of the Second European Conference on Computational Learning Theory*, EuroCOLT '95, Springer-Verlag, London, UK, UK, pp. 23–37.

Fu, Z. and Han, Y. (2012). Centroid weighted kalman filter for visual object tracking, *Measument* **46**: 250–655.

Gabor, D. (1946). Theory of communication, *J. Inst. Elect. Eng.* **93**: 429–457.

Gong, S., McKenna, S. and Psarrou, A. (2000). *Dynamic Vision: From Images to Face Recognition*, Imperial College Press.

Gonzalez, R. C. and Woods, R. E. (2006). *Digital Image Processing (3rd Edition)*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

He, C., Zheng, Y. F. and Ahalt, S. C. (2002). Object tracking using the gabor wavelet transform and the golden section algorithm, *Trans. Multi.* **4**(4): 528–538.

Hotta, K. (2009). Adaptive weighting of local classifiers by particle filters for robust tracking, *Pattern Recognition* **42**(5): 619–628.

Huang, F. J. and Chen, T. (2000). Tracking of multiple faces for human-computer interfaces and virtual environments, *Proc. IEEE Int. Conf. Multimedia and Expo ICME 2000*, Vol. 3, pp. 1563–1566.

Jepson, A., Fleet, D. and El-Maraghi, T. (2003). Robust online appearence models for visual tracking, *Pattern Analysis and Machine Inteligence* **25**: 1296–1311.

Karavasilis, V., Nikou, C. and Likas, A. (2011). Visual tracking using the earth mover's distance between gaussian mixtures and kalman filtering, *Image Vision Comput.* **29**(5): 295–305.

Kim, M., Kumar, S. and Pavlovic, V. (2008). Face tracking and recognition with visual constraints in real-world videos, *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, IEEE, pp. 1–8.

Lades, M., Vorbruggen, J., Buhmann, J., Lange, J., von der Malsburg, C., Wurtz, R. and Konen, W. (1993). Distortion invariant object recognition in the dynamic link architecture, *Computers, IEEE Transactions on* **42**(3): 300 –311.

Lee, B. Y., Liew, L. H., Cheah, W. S. and Wang, Y. C. (2012). Measuring the effects of occlusion on kernel based object tracking using simulated videos, *Procedia Engeneering: International Symposium on Robotics and Intelligent Sensors 2012*, Vol. 41, Elsevier, pp. 764–770.

Lee, H.-S. and Kim, D. (2007). Robust face tracking by integration of two separate trackers: Skin color and facial shape, *Pattern Recogn.* **40**(11): 3225–3235.

Lee, K. C., Ho, J., Yang, M. H. and Kriegman, D. (2005). Visual tracking and recognition using probabilistic appearance manifolds, *Computer Vision and Image Understanding* pp. 313–320.

Lee, K., Ho, J. and Kriegman, D. (2005). Acquiring linear subspaces for face recognition under variable lighting, *IEEE Trans. Pattern Anal. Mach. Intelligence* **27**(5): 684–698.

Lee, T. S. (1996). Image representation using 2d gabor wavelets, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**: 959–971.

Li, S. Z. and Jain, A. K. (2005). *Handbook of Face Recognition*, Springer.

Lin, C. (2007). Face detection in complicated backgrounds and different illumination conditions by using ycbcr color space and neural network, *Pattern Recogn. Lett.* **28**(16): 2190–2200.

Liu, Z., Shen, H., Feng, G. and Hu, D. (2012). Tracking objects using shape context matching, *Neurocomputing* **83**(0): 47 – 55.

Mallat, S. (2008). *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*, 3rd edn, Academic Press.

Nummiaro, K., Koller-Meier, E. and Gool, L. V. (2002). An adaptive color-based particle filter, *Image and Vision Computing* **3**(1): 99–110.

Osuna, E., Freund, R. and Girosi, F. (1997). Support vector machines: Training and applications, *Technical report*, Cambridge, MA, USA.

Pankanti, S., Bolle, R. M. and Jain, A. (2000). *Biometrics: The future of identification*, Computer.

Peer, P., Kovac, J. and Solina, F. (2003). Human skin colour clustering for face detection, *Proceedings of the International Conference on Computer as a Tool EUROCON 2003*, Vol. 2, IEEE, pp. 144–148.

Pentland, A. and Choudhury, T. (2000). Face recognition for smart environments, *Computer* **33**(2): 50–55.

Phillips, P. J., Wechsler, H., Huang, J. and Rauss, P. (1998). The FERET database and evaluation procedure for face recognition algorithms, *Image and Vision Computing* **16**(5): 295–306.

Platt, J. C. (1999). Advances in kernel methods, MIT Press, Cambridge, MA, USA, chapter Fast training of support vector machines using sequential minimal optimization, pp. 185–208.

Ratha, N. K., Senior, A. W. and Bolle, R. M. (2001). Automated biometrics, *Proceedings of the Second International Conference on Advances in Pattern Recognition*, ICAPR '01, Springer-Verlag, London, UK, UK, pp. 445–474.

Ross, D. A., Lim, J., Lin, R.-S. and Yang, M.-H. (2008). Incremental learning for robust visual tracking, *Int. J. Comput. Vision* **77**(1-3): 125–141.

Rowley, H. A., Member, S., Baluja, S. and Kanade, T. (1998). Neural network-based face detection, *IEEE Transactions On Pattern Analysis and Machine intelligence* **20**: 23–38.

Schölkopf, B., Smola, A. J., Williamson, R. C. and Bartlett, P. L. (2000). New support vector algorithms, *Neural Comput.* **12**(5): 1207–1245.
URL: *http://dx.doi.org/10.1162/089976600300015565*

Seo, N. (2009). *Simultaneous multi-view face racking and recognition in video using particle filtering*, Master's thesis, University of Maryland.

Serrano, A., Martín de Diego, I., Conde, C. and Cabello, E. (2011). Analysis of variance of gabor filter banks parameters for optimal face recognition, *Pattern Recogn. Lett.* **32**(15): 1998–2008.

Shen, L., Bai, L. and Fairhurst, M. (2007). Gabor wavelets and general discriminant analysis for face identification and verification, *Image and Vision Computing* **25**(5): 553 – 563.

Sung, K.-K. and Poggio, T. (1998). Example-based learning for view-based human face detection, *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(1): 39–51.

Terrillon, J.-C., Pilpr, A., Niwa, Y. and Yamamoto, K. (2004). Druide : A real-time system for robust multiple face detection, tracking and hand posture recognition in color video sequences, *Pattern Recognition, International Conference on* **3**: 302–305.

Thomaz, C. E. and Giraldi, G. A. (2010). A new ranking method for principal components analysis and its application to face image analysis, *Image Vision Comput.* **28**(6): 902–913.

Toyama, K. (1998). Prolegomena for robust face tracking, *Technical Report 14*, Microsoft Research.

Toyama, K. and Hager, G. D. (1997). If at first you don't succeed..., *in Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, Providence, Rhode Island, USA, pp. 3–9.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition, *J. Cognitive Neuroscience* **3**(1): 71–86.

Vapnik, V. N. (1995). *The nature of statistical learning theory*, Springer-Verlag New York, Inc., New York, NY, USA.

Viola, P. A. and Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features, *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, Vol. 1, pp. 511–518.

Yang, H., Shao, L., Zheng, F., Wang, L. and Song, Z. (2011). Recent advances and trends in visual tracking: A review, *Neurocomputing* **74**(18): 3823 – 3831.

Yang, M., Zhang, L., Shiu, S. C. and Zhang, D. (2012). Gabor feature based robust representation and classification for face recognition with gabor occlusion dictionary, *Pattern Recognition* (0): –. In Press, Corrected Proof.
URL: *http://www.sciencedirect.com/science/article/pii/S0031320312002920*

Yanushkevich, S. N. (2006). Synthetic biometrics: A survey, *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2006, part of the IEEE World Congress on Computational Intelligence, WCCI 2006, Vancouver, BC, Canada, 16-21 July 2006*, IEEE, pp. 676–683.

Yao, H. and Gao, W. (2000). Face locating and tracking method based on chroma transform in color images, *Proc. 5th Int. Conf. Signal Processing WCCC-ICSP 2000*, Vol. 2, pp. 1367–1371.

Yilmaz, A., Javed, O. and Shah, M. (2006). Object tracking: A survey, *ACM Comput. Surv.* **38**(4): 1–45.

Yin, S., Na, J. H., Choi, J. Y. and Oh, S. (2011). Hierarchical kalman-particle filter with adaptation to motion changes for object tracking, *Comput. Vis. Image Underst.* **115**(6): 885–900.

Zhang, K., Zhang, L. and Yang, M.-H. (2012). Real-time compressive tracking, *Proceedings of the 12th European conference on Computer Vision - Volume Part III*, ECCV'12, Springer-Verlag, Berlin, Heidelberg, pp. 864–877.

Zhao, W., Chellappa, R. and Phillips, P. J. (2003). Face recognition: A literature survey, *ACM Computing Surveys* **35**(4): 399–458.

Zhou, S., Chellappa, R. and Krueger, V. (2002). Probabilistic recognition of human faces from video, *Computer Vision and Image Understanding* **91**: 214–245.

Apêndice A

Produção Bibliográfica Associada à Tese

Durante o desenvolvimento da pesquisa para elaborar esta Tese de Doutorado foram publicados os seguintes trabalhos científicos:

- Cornelia Janayna Pereira Passarinho, Evandro Ottoni Teattini Salles, Mario Sarcinelli Filho, On Face Tracking in Video Sequences, IWSSIP 2010 - 17th International Conference on Systems, Signals and Image Processing, Niterói, RJ, Brasil, 2010, pp. 308-311.
- Cornelia Janayna Pereira Passarinho, Evandro Ottoni Teattini Salles, Mario Sarcinelli Filho, On face Face Tracking in Unconstrained Videos Sequences, XVIII Congresso Brasileiro de Automática - CBA2010, Bonito, MS, 2010, pp. 130-135.
- Cornelia Janayna Pereira Passarinho, Evandro Ottoni Teattini Salles, Mario Sarcinelli Filho, Face Detection Based on Adaptive Support Vector Tracker, ISSNIP Biosignals and Biorobotics Conference 2011, Vitória, ES, Brasil, 2011.
- Cornelia Janayna Pereira Passarinho, Evandro Ottoni Teattini Salles, Mario Sarcinelli Filho, Detection and Tracking Faces in Unconstrained Color Video Streams, 7th International Symposium on Visual Computing, 2011, Las Vegas. Lecture Notes in Computer Science, 2011, v. 6939, pp. 466-475.
- Cornelia Janayna Pereira Passarinho, Evandro Ottoni Teattini Salles, Mario Sarcinelli Filho, Face Tracking Framework Using Face Detection in Color Image Multi View with Multi Skin, XIX Congresso Brasileiro de Automática (CBA 2012), Campina Grande, PB, 2012, v. 1, p. 4057-4063.

- Cornelia Janayna Pereira Passarinho, Evandro Ottoni Teattini Salles, Mario Sarcinelli Filho, Detecting and Tracking a Face in Unconstrained Color Video Streams, submetido em 19/10/2012 ao periódico Image and Vision Computing, ISSN 0262-8856, sob o número IMAVIS-D-12-00539 (fator de impacto JCR 1,743, Qualis CAPES A1 pelo Comitê Engenharias IV).

Apêndice B

Código Correspondente ao Seguidor SDVS

Abaixo está listado o código completo correspondente ao seguidor SDVS aqui proposto, escrito em $MATLAB^{®}$:

```
% SDVS principal
%
mov = lersequenciavideo('video.avi');
inicio = 1;
fim = length(mov);
estado_estimado(2) = 0;
estado_estimado(1) = 0;
tipo = 'RMO';
obsface = zeros(10,2);
centros = zeros(10,2);
xvpredito = zeros(2,1);
xvpredito(2,1)= 0;
janela = 35;
ant = [0,0];
ss = 4;
Variancia_pred_ant = 5*eye(ss);
Variancia_estimada = 5*eye(ss);
centros(inicio,:) = [coordenadasX,coordenadasY];
obsface(inicio,:) = [coordenadasX,coordenadasY];
```

```
observacao(2)= coordenadasX;
observacao(1)= coordenadasY;
coordenadasX_ant = 0;
coordenadasY_ant = 0;
numoclusoesSeguidas = 0;
temp_tracking = [];
 for i=inicio+1:fim
    frame = mov(i).cdata;
    tic;
    numoclusoesSeguidas = 0;
        coordenadas = [ceil(estado_estimado(2)-janela),ceil(estado_estimado(2)+jane
        Cfinal = imag_improve_rgb(frame);
        resultadopele = faceexantigo(Cfinal);
        im2 = Cfinal(:,:,2);
        [numerodefaces,coordenadasX,coordenadasY,cantos] = testaSVM(im2,model,resul
        if oclusao==0
            % predição da localização da face para o proximo frame  usando
            % filtro de Kalman
            observacao(2)= coordenadasX;
            observacao(1)= coordenadasY;
            [observacao_estimada,estado_estimado,Variancia_estimada] = segfacekalma
            coordenadasX_ant = estado_estimado(2);
            coordenadasY_ant = estado_estimado(1);
            Variancia_pred_ant = Variancia_estimada;
            oclusao = 0;
        else
            oclusao = 1;
            numoclusoesSeguidas = numoclusoesSeguidas +1;
            [observacao_estimada,estado_estimado,Variancia_estimada] = segfacekalma
            observacao(2)= 0;
            observacao(1)= 0;
            coordenadasX_ant = estado_estimado(2);
            coordenadasY_ant = estado_estimado(1);
        end

    temp_tracking(i) = toc;
    centros(i,:) = [estado_estimado(1),estado_estimado(2)];
    obsface(i,:) = [coordenadasX,coordenadasY];
```

```
    figure(1),imshow(frame),title(num2str(i));
    hold on;
       rectangle('Position',[estado_estimado(1)-15,estado_estimado(2)-20,janela,jan
        plot(estado_estimado(1),estado_estimado(2),'b*');
    hold off;
    pause(0.1);

 end

% Varre a janela de busca por faces
function [numerodefaces,coordenadaX,coordenadaY,cantos] = testaSVM(im2,model,result
[RespostaMaximaOriginal,do,X] = GaborOriginal(im2,'conv');
[lf,cf]=size(im2);
lf = coordenadas(2);
cf = coordenadas(4);
quantasfaces=0;
tambloco1 = 40;
tambloco2 = 30;
cont=0;
valordec = 0;
% valordecN = 0;
coordenadascb = [];
coordenadaslb = [];
coordenadasX = [];
coordenadasY = [];
rotulo_pre=0;
for lb=coordenadas(1):lf-tambloco1
    for cb=coordenadas(3):cf-tambloco2
        y = lb;
        x = cb;
        if (cb<0)
            x = 1;
        end
        if (lb<0)
            y = 1;
        end
        if (y <= lf-tambloco1-1) && (x <= cf-tambloco2-1)
```

```
if (resultado(y+1,x+1)==1)
    if y+tambloco1> lf
        y = lf - tambloco1;
    end

    if x+tambloco2 > cf
        x = cf - tambloco2;
    end
    cj=im2(y+1:y+tambloco1,x+1:x+tambloco2);
    if strcmp(tipo,'X')
        auxface = (X{1});
        auxface = auxface(y+1:y+tambloco1,x+1:x+tambloco2);
        auxface = auxface(: )';
        dadosface = auxface;
        auxface = X{2};
        auxface = auxface(y+1:y+tambloco1,x+1:x+tambloco2);
        auxface = auxface(: )';
        dadosface = [dadosface,auxface];
        auxface = (X{3});
        auxface = auxface(y+1:y+tambloco1,x+1:x+tambloco2);
        auxface = auxface(: )';
        dadosface = [dadosface,auxface];
        auxface = (X{4});
        auxface = auxface(y+1:y+tambloco1,x+1:x+tambloco2);
        auxface = auxface(: )';
        dadosface = [dadosface,auxface];
    end
    if strcmp(tipo,'RMO')
        auxface = RespostaMaximaOriginal;
        auxface = auxface(y+1:y+tambloco1,x+1:x+tambloco2);
        dadosface = auxface(: )';
    end
    if strcmp(tipo,'RMOX')
        auxface = abs(X{1});
        auxface = auxface(y+1:y+tambloco1,x+1:x+tambloco2);
        auxface = auxface(: )';
        dadosface = auxface;
        auxface = X{2};
```

```
                    auxface = auxface(y+1:y+tambloco1,x+1:x+tambloco2);
                    auxface = auxface(: )';
                    dadosface = [dadosface,auxface];
                    auxface = abs(X{3});
                    auxface = auxface(y+1:y+tambloco1,x+1:x+tambloco2);
                    auxface = auxface(: )';
                    dadosface = [dadosface,auxface];
                    auxface = abs(X{4});
                    auxface = auxface(y+1:y+tambloco1,x+1:x+tambloco2);
                    auxface = auxface(: )';
                    dadosface = [dadosface,auxface];
                    auxface = RespostaMaximaOriginal;
                    auxface = auxface(y+1:y+tambloco1,x+1:x+tambloco2);
                    auxface = auxface(: )';
                    dadosface = [dadosface,auxface];
                end
                [rotulo_pre, acuracia, valordecimal] = svmpredict(2, double(dadosfa
                if  rotulo_pre >0
                   coordenadascb(quantasfaces+1) = cb;
                   coordenadaslb(quantasfaces+1) = lb;
                   face = im2(y+1:y+tambloco1,x+1:x+tambloco2);
                   imauxteste(y+1:y+tambloco1,x+1:x+tambloco2) = 1;
                   [xalvo,yalvo] = find(imauxteste==1);
                   valordec(quantasfaces+1) = valordecimal;
                   coordenadasX(quantasfaces+1) = sum(xalvo)/length(xalvo);
                   coordenadasY(quantasfaces+1) = sum(yalvo)/length(yalvo);
                   quantasfaces = quantasfaces+1;
               end
            end
        end
    end
end

numerodefaces = quantasfaces;
[val,loc] = min(valordec);

 if numerodefaces>0
        figure(1),imshow(im3);title(strcat(num2str(numframe),' - ',num2str(val)));
```

```
         hold on;
            rectangle('Position',[coordenadascb(loc)+1,coordenadaslb(loc)+1,tambloc
         hold off;
end
    coordenadaX = 0;
    coordenadaY = 0;
    cantos = [estadoestimado(2),estadoestimado(1)];
   if numerodefaces==1
    coordenadaX = coordenadasX(loc);
    coordenadaY = coordenadasY(loc);
    cantos = [coordenadascb(loc)-tambloco2,coordenadaslb(loc)-tambloco1];
   end
    if numerodefaces>1
    coordenadaX = mean(coordenadasX(1:2));
    coordenadaY = mean(coordenadasY(1:2));
    coordenadascb = mean(coordenadascb(1:2));
    coordenadaslb = mean(coordenadaslb(1:2));
    cantos = [coordenadascb,coordenadaslb];
    end
end

% procurando o C e gama que oferecem melhores resultados

matlabpool(4)
end

bestcv = 0;

tic
parfor log2c = -15:5,
    for log2g = -15:5,
        ii = log2c;
        jj = log2g;
        disp (['Executanto ' num2str(log2c) ' e ' num2str(log2g)])
        op = ['-t 2 -v 5 -c ', num2str(2^log2c), ' -g ', num2str(2^log2g), '-h 0'];
        cv = svmtrain(rot,dado,op); % essa matriz vazia é para não utilizar pesos n

        fid = fopen(['Arq_log2c_' num2str(ii) '_log2g_' num2str(jj) '.txt'],'w');
```

```
        fprintf(fid,'%g %g %g\n', ii, jj, cv);
        fclose(fid);

        fprintf('%g %g %g\n', log2c, log2g, cv); %, bestc, bestg, bestcv);
    end
end

toc

% Procurando os melhores valores para C e gama e treinamento do SVM com kernel RBF

inicio = -15;
fim = 5;

x = [];
cv = 0;
for log2c = inicio:fim,
    for log2g = inicio:fim,
        ii = log2c - inicio + 1;
        jj = log2g - inicio + 1;

        dado = load(['Arq_log2c_' num2str(log2c) '_log2g_' num2str(log2g) '.txt']);
        Z(ii,jj) = dado(3);

        if cv < dado(3)
            cv = dado(3);
            ind = [ii jj log2c log2g cv];
        end

    end
end

[X,Y] = meshgrid(inicio:fim);
mesh(X,Y,Z)
% colormap(gray);
xlabel('log2(g)')
ylabel('log2(C)')
zlabel('cv')
```

```
disp('Resultado de máximo CV')
disp(['log2c = ' num2str(ind(3)) ', log2g = ' num2str(ind(4)) ', cv = ' num2str(ind
disp(' ')
disp('Treinar e salvar modelo: Pressione ENTER')
pause

disp('Treinando o SVM com os parametros escolhidos');
bestc = 2^ind(3);
bestg = 2^ind(4);

load dadostreinonovembro; % carrega os dados;
dado = (dados - min(dados(:)))/(max(dados(:)) -min(dados(:)));
op = ['-t 2 -c ', num2str(2^ind(3)), ' -g ', num2str(2^ind(4)), '-h 0'];

model = svmtrain(double(rot),double(dado), op);

% save model;
disp('Fim do Treinamento.');


% Compensação de iluminação

function Jteste = imag_improve_rgb(IMG)
% figure,imshow(IMG)
% title('original'),pause;
R=double(IMG(:,:,1));
G=double(IMG(:,:,2));
B=double(IMG(:,:,3));
[H,W]=size(R);

minR=min(R(:));
minG=min(G(:));
minB=min(B(:));

[srow,scol]=find(R==0 & G==0 & B==0);
if(isempty(srow) && isempty(scol))
    minR=min(R(:));
```

```
    minG=min(G(:));
    minB=min(B(:));
end
R=R-minR;
G=G-minG;
B=B-minB;

S=zeros(H,W);
[srow,scol] = find(R==0 & G==0 & B==0);
[sm,sn]=size(srow);
for i=1:sm
     S(srow(i),scol(i))=1;
end
mstd=sum(S(:));
Nstd=(H*W)-mstd;

Cst=0;
Cst=double(Cst);
for i=1:H
    for j=1:W
         a=R(i,j);
         b=R(i,j);

            if(B(i,j)<a)
               a=B(i,j);
            else
               b=B(i,j);
            end

             if(G(i,j)<a)
                a=G(i,j);
             else
                b=G(i,j);
             end

         Cst=a+b+Cst;

    end
```

```
end
Cstd = Cst/(2*Nstd);
CavgR=sum(R(:))./(H*W);
CavgB=sum(B(:))./(H*W);
CavgG=sum(G(:))./(H*W);
Rsc=Cstd./CavgR;
Gsc=Cstd./CavgG;
Bsc=Cstd/CavgB;
R=R.*Rsc;
G=G.*Gsc;
B=B.*Bsc;
C(:,:,1)=R;
C(:,:,2)=G;
C(:,:,3)=B;
Jteste =C;

% Detecção dos pixel com tons de pele

function final_image = faceE(img)
final_image = zeros(size(img,1), size(img,2));
if(size(img, 3) > 1)
    for i = 1:size(img,1)
        for j = 1:size(img,2)
            R = img(i,j,1);
            G = img(i,j,2);
            B = img(i,j,3);
            if(R > 20 && G > 30 && B < 100)
                v = [R,G,B];
                    if( R > G && R > B)
                        %it is a skin
                        final_image(i,j) = 1;
                    end
            end
        end
    end
end

% Filtro de Kalman
```

```
function [observacao_estimada,estado_estimado,Variancia_estimada] = segfacekalman(c

% Dinâmica do sistema
% x = A*y  + v;
% y = C*y + w;

%posição do centro de massa da face.
[l,c] = size(coordenadasX);
estado_anterior(2) = coordenadasX;
estado_anterior(1) = coordenadasY;
estado_anterior(3) = 1;
estado_anterior(4) = 0;
estado_anterior = estado_anterior(:);
observacao = observacao(:);

%%%%%%%%%%%%%%%%%%%%% Kalman filtering %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A = [1 0 1 0;0 1 0 1;0 0 1 0;0 0 0 1];
C = [1 0 0 0;0 1 0 0];
% ss = 4;
ss = size(A,2);
os = 2;
Q = 1*eye(ss);
R = 1*eye(os);

% predição
    mu_predito = A*estado_anterior;
    Variancia_pred = A*Variancia_pred_ant*A' + Q;
    % ganho de Kalman
    Kk = Variancia_pred*C'*inv(C*Variancia_pred*C'+R);
    % correção
    if (oclusao==0)
        mu_atual = mu_predito+ Kk*(observacao - C*mu_predito);
        Variancia_atual = (eye(ss)-Kk*C)*Variancia_pred;
    else
        mu_atual = mu_predito;
        Variancia_atual = Variancia_pred;
```

```
    end

    %
    observacao_estimada = C*mu_predito;
    estado_estimado = mu_atual;
    Variancia_estimada = Variancia_atual;

% Características de Gabor

function [RespostaMaximaOriginal,do,X] = GaborOriginal(imagem,op);

% Caracteristicas de gabor
% close all;
%tamanho da janela Gaussiana
N = 9;

frec = 0.25;
ro1 = 2; ro2 = sqrt(ro1);
res = 1;
[l,c]=size(imagem);

orientacaoQuatro = [pi/4 pi/2 3*pi/4 pi];
X = cell(1,length(orientacaoQuatro));
h = cell(1,length(orientacaoQuatro));
RespostaMaximaOriginal = 0;
for oQ=1:length(orientacaoQuatro)
 if strcmp(op,'conv')
    h{oQ} = gabor( N, N, orientacaoQuatro(oQ), frec, ro1, ro2);
    X{oQ} = imfilter(imagem,real(h{oQ}),'symmetric');
 else
    h{oQ} = gabor( c, l, orientacaoQuatro(oQ), frec, ro1, ro2);
    imagem_fft = fft2(imagem);
    imagem_fft = fftshift(imagem_fft);
    h{oQ}  = fft2(h{oQ});
    h{oQ}  = fftshift(h{oQ});
    X{oQ} = h{oQ}.*imagem_fft;
    X{oQ} = fftshift(ifft2((X{oQ})));
 end
```

```
    RespostaMaximaOriginal = max(abs(RespostaMaximaOriginal),abs(X{oQ}));
end
```