Detecção e rastreamento de faces e características
faciais com aplicação em indexação de vídeo
e estimação da orientação de faces

Regis de Abreu Barbosa

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DE TÍTULO DE MESTRE
EM
CIÊNCIAS

Área de Concentração: Ciência da Computação
Orientador: Prof. Dr. Carlos Hitoshi Morimoto

Durante a elaboração deste trabalho o autor recebeu auxílio financeiro da FAPESP

São Paulo, agosto de 2006.

Agradecimentos

Agradeço, antes de tudo, a Deus, que tem sempre se mostrado presente em minha vida e me permitiu alcançar mais esta vitória.

Agradeço aos meus pais, irmãos e familiares, que me sustentaram, me apoiaram e me incentivaram a sempre continuar firme nos meus estudos.

Agradeço ao meu orientador pela dedicação e pelas longas conversas sobre este trabalho e sobre a minha vida profissional.

Agradeço à FAPESP pela bolsa concedida, aos meus amigos da USP, aos amigos que conheci ao participar de congressos da Sibgrapi em Curitiba e Natal.

E agradeço a minha amada esposa Carolina pela cooperação, carinho e por me fazer mais feliz a cada dia.

A todos, muito obrigado.

Resumo

A detecção e o rastreamento de faces e características faciais continuam sendo grandes desafios da área de Visão Computacional, especialmente para serem implementados em tempo real, requisito necessário para uma interação natural entre o homem e o computador. Nesse trabalho, desenvolvemos métodos para detecção e rastreamento de faces e características faciais apropriados para aplicações interativas. Nosso sistema utiliza um modelo de cor de pele baseado no espaço TSL para detectar possíveis regiões de face na imagem. A estas regiões aplicamos um algoritmo para localizar candidatos a olhos, que são então pareados baseados em regras heurísticas, e cada par define então uma região candidata a face. Para determinar se um candidato é realmente uma face, utilizamos um método baseado em análise de componentes principais. Outras características da face, como boca e nariz, são localizadas através de correlação. Para eliminar falsos candidatos, utilizamos modelos deformáveis baseados na técnica de Modelos de Forma Ativa (*Active Shape Model*) para determinar se as características localizadas estão de acordo com um modelo pré-determinado.

Uma vez detectada, a face é rastreada de forma bastante eficiente através do método CAMShift. Este método utiliza o histograma de uma componente cromática do objeto e a sua posição anterior para determinar sua posição no quadro atual. Implementamos também métodos baseados em correlação para rastrear características faciais, como olhos, nariz e boca.

Estes algoritmos foram utilizados no desenvolvimento de dois aplicativos. O primeiro é uma ferramenta automática de indexação de vídeo que permite organizar e realizar consultas a um repositório de dados de vídeo baseado no conteúdo das imagens. O segundo aplicativo é um estimador da orientação de face, que se utiliza do algoritmo POS/POSIT.

Estes aplicativos se mostraram mais precisos quando treinados para usuários específicos. Porém quando os parâmetros desses usuários são utilizados para outros, a precisão do sistema cai consideravelmente. Mesmo assim, o sistema pode ser utilizado em aplicações específicas que não exigem grande precisão, como indexação de vídeo e interação com computadores.

Abstract

Face and facial features detection and tracking are still great challenges, specially to be implemented in real time, which is required to a natural interaction between humans and computers. In this work, we developed methods to detect and track faces and facial features applied to interactive applications. We chose a method based on a skin color model using TSL color space to detect possible face regions in the image. In these regions, we apply an algorithm to detect the eyes candidates that are paired based on some heuristic rules, and each pair of eyes define a candidate region to be a face. To determine if a candidate is really a face, we apply a method based on PCA - Principal Component Analysis. The other facial features are localized using correlation. To eliminate false face candidates, we use the Active Shape Model technique to determine whether the localized features agree with a face model.

Once we have the face, we can track it using an efficient method called CAMShift. It uses the histogram of a chromatic component of the object and its position in the last frame to determine its position in the current frame. The facial features as eyes, nose and mouth are also tracked through methods based on correlation.

We show two examples of practical applications of the implemented algorithms. The first is an Automatic Video Browsing Tool that allows queries to a video data repository based on the content of the images. The second one is a head pose estimator, that uses The Pos/posit algorithm.

These applications have more precision when trained for specific users. However when the parameters of these users are used for others, the precision of the system decreases. Despite this, the system can be used in specific applications that do not demand great precision such as video indexing and interaction with computers.

Sumário

1	Introdução	1
1.1	Motivação	3
1.2	Objetivo	4
1.3	Contribuições	4
1.4	Organização da Dissertação	6
2	Revisão Bibliográfica	7
2.1	Detecção de faces	9
2.1.1	Métodos baseados em conhecimento	9
2.1.2	Métodos baseados em características invariantes	9
2.1.3	Métodos baseados em ajustes de modelos de forma	13
2.1.4	Métodos baseados em aparência	13
2.2	Rastreamento de faces	14
3	Detecção e Rastreamento de Faces e Características Faciais	15
3.1	Detecção de Cor de Pele	16
3.2	Análise de Componentes Principais	21
3.3	Detecção de Características Faciais	26

3.3.1	Detecção de olhos pelo gradiente	28
3.3.2	Detecção de características por correlação	39
3.4	Modelos Deformáveis	42
3.5	Rastreamento de faces e características faciais	50
4	Resultados Experimentais e Aplicações	54
4.1	Resultados da detecção de cor de pele	54
4.2	Resultados da detecção de características	56
4.3	Aplicações	58
4.3.1	Indexação em vídeo	58
4.3.2	Estimação da Orientação da Face	62
5	Comentários finais	77

Lista de Figuras

2.1	Fontes de luz infra-vermelha L1 e L2.	11
2.2	Imagens com as pupilas brilhantes e escuras e a diferença entre as imagens. . .	12
3.1	Diagrama do sistema de detecção de faces e características faciais.	16
3.2	Extração de regiões de cor de pele para conjunto de treinamento.	18
3.3	Resultado do cálculo da probabilidade de cor de pele para cada pixel.	19
3.4	Exemplo de histograma particionado em intervalos de 10/100.	20
3.5	Resultado da limiarização aplicada ao exemplo dado na figura 3.3.	21
3.6	Resultado da limiarização seguida de operações para eliminação de ruídos. . .	22
3.7	Imagens de treinamento de PCA (ao alto) e face média.	26
3.8	Resultado da aplicação do filtro de Sobel na direção x (à esquerda), y (à direita) e a magnitude (abaixo).	32
3.9	Resultado da limiarização sobre o resultado do filtro de Sobel com limiares de 20 (à esquerda), 30(à direita) e 40 (abaixo).	33
3.10	Resultado do algoritmo de Canny para extração de bordas.	34
3.11	(a) Matriz de acumulação para raios entre 6 e 9, (b) Pontos candidatos a olhos em azul e par de olhos escolhido pelo fator J (vide seção 3.3.1) em amarelo. . .	37

3.12	(a) Matriz de acumulação criada pelo algoritmo de detecção pela direção do gradiente, (b) Pontos candidatos a olhos em azul e par de olhos escolhido pelo fator J em amarelo.	38
3.13	Exemplos das características faciais utilizadas para cálculo da correlação. . . .	40
3.14	Exemplo de aplicação de correlação. A primeira linha mostra a janela $f(x, y)$, a segunda a amostra $h(x, y)$, a terceira o resultado da correlação e a quarta mostra a posição da característica.	41
3.15	(a) Regiões de busca de características por correlação, (b) Resultado da busca.	42
3.16	Exemplo de erro da busca por correlação.	43
3.17	Iterações do algoritmo de busca por ASM.	49
3.18	CAMShift	52
3.19	Rastreamento com CAMShift	53
4.1	Imagens de face utilizadas para treinamento de cor de pele.	55
4.2	Resultados da detecção de cor de pele	56
4.3	Resultados da detecção de características faciais utilizando o algoritmo de detecção pelo gradiente para detecção de olhos e correlação para as demais características.	57
4.4	Interface da ferramenta de indexação em vídeo.	60
4.5	Sistema de Coordenadas da Câmera e Objeto [DD95]	64
4.6	Face com orientação de 58° no plano vertical e 58° no plano horizontal.	69
4.7	Exemplos de entrada.	70
4.8	Média de acertos da detecção de características faciais	72
4.9	Exemplos de saída da estimação de orientação da face.	75
4.10	Exemplos de saída da estimação de orientação da face de uma mesma pessoa. .	76

Capítulo 1

Introdução

Os avanços na tecnologia de imageamento digital tornaram acessível à população o uso desta tecnologia. O que antes era uma simples câmera fotográfica tornou-se uma máquina que, aliada a um computador, é capaz de criar as mais diferentes imagens digitais, melhorando sua qualidade ou mesmo alterando esta imagem, de acordo com a criatividade do usuário.

A Internet expandiu ainda mais o uso da imagem digital. Sistemas de monitoramento e segurança passaram a integrar imagens em seus bancos de dados como um novo recurso para identificação de pessoas. Câmeras digitais passaram a ser utilizadas para identificar o acesso de pessoas a áreas restritas. Os sistemas de comunicação de tempo real passaram a integrar imagens de vídeo, além da voz e mensagens de texto, oferecendo às pessoas outros recursos ainda inexistentes.

O aumento da capacidade de armazenamento de dados dos computadores aliado a técnicas de compressão de imagens permitiu a criação de bancos de imagens e vídeos digitais, que podem servir a várias aplicações específicas ou funcionar como uma biblioteca multimídia. Mas, enquanto a imagem digital é uma grande fonte de informação e entretenimento, pouco desta informação é realmente aproveitada por programas de computador, pela falta de ferramentas

que permitam buscar, organizar e indexar informações visuais.

A Visão Computacional é uma área de pesquisa que surgiu da Inteligência Artificial com a finalidade de entender cenas, ou seja, os objetos que compõem o ambiente que nos cerca. Da pesquisa em visão computacional surgiram diversas aplicações relacionadas à análise do conteúdo de imagens, sejam estas imagens médicas, imagens de ambientes industriais ou imagens de faces. A análise de imagens de faces, objeto de nosso estudo, encontra aplicações em especial na área de Interação Humano-Computador, gerando um novo canal de comunicação entre o homem e a máquina. A posição de uma face bem como sua orientação trouxe ao computador uma informação preciosa para que este interagisse com o usuário de uma forma mais natural. As expressões faciais do usuário possibilitam um conhecimento maior de sua satisfação ou reação a determinados eventos. A identificação dos usuários permite ainda um comportamento personalizado do computador na sua interação com indivíduos distintos.

A análise de faces encontra também aplicações em áreas de realidade virtual, como mundos virtuais interativos, jogos, teleconferência; em aplicações relacionadas à segurança como sistemas inteligentes de vigilância e sistemas de controle de acesso; e em modelagem de faces como codificação baseada em modelo para compressão de vídeo e animação de modelos de faces.

Mas esta não é uma tarefa trivial. A grande variação de formas, cores de faces, presença de barba, óculos, variações de orientação e expressão facial e variações de iluminação da imagem tornam a definição de um modelo capaz de representar a face de forma generalizada um grande desafio.

Atualmente já existem muitos sistemas de análise de face, mas a maioria requer algum tipo de sensor conectado à cabeça do usuário. Os novos estudos têm se concentrado na execução da tarefa sem necessidade do usuário “vestir” nenhum componente de hardware, permitindo uma interação mais natural com esses sistemas computacionais.

1.1 Motivação

Desenvolvemos esse trabalho dentro do Laboratório de Tecnologias para Interação (LaTIn) do IME-USP, que tem se concentrado no desenvolvimento de novas interfaces e técnicas de interação com o computador, como, por exemplo, técnicas para rastreamento de olhar [ZMI99, Bar05] e interfaces baseadas em gestos [Cab05].

Qualquer sistema de análise de faces requer como entrada a imagem de uma face. Logo, uma tarefa primordial para estes sistemas consiste na detecção de faces na imagem.

Embora seja muito natural para a visão humana, a detecção de faces e características faciais é um grande desafio para sistemas de visão computacional.

Podemos definir a detecção de faces da seguinte maneira: dada uma imagem, o objetivo desta tarefa é determinar se existe ou não uma face na imagem. Em caso positivo, retornar a sua localização. Semelhantemente, a detecção de características faciais é responsável por detectar a presença e localizar características como olhos, sobrancelhas, nariz, narinas, boca, lábios, ouvidos, etc.

Alguns fatores interferem diretamente na detecção de face e características faciais, como a orientação da face, as expressões faciais, a oclusão parcial da face e a presença ou ausência de componentes estruturais como barba, bigode, óculos, maquiagem, etc. Todos estes são desafios a serem enfrentados por um algoritmo de detecção de face.

Além de detectar a face, outro problema abordado nesta pesquisa é o seu rastreamento, que pode ser definido como a localização contínua de uma mesma face em uma sequência de vídeo, ou seja, não basta simplesmente aplicarmos um algoritmo de detecção de face nos quadros consecutivos de uma sequência de vídeo, pois precisamos nos certificar de que as faces detectadas correspondem ao mesmo indivíduo. Além disso, para aplicação do rastreamento de face em sistemas interativos, precisamos que esta tarefa seja realizada em tempo real.

1.2 Objetivo

O objetivo de nossa pesquisa é criar um dispositivo para aplicações em ambientes interativos. A posição da face e de outras características faciais nos permite obter informações sobre a sua orientação e expressão facial, que podem ser utilizadas por diversos outros sistemas computacionais para fins de interação, e podem ser também aplicadas, por exemplo, para indexação e segmentação de vídeo digital.

Neste trabalho, buscamos construir um sistema capaz de localizar faces frontais em imagens e rastreá-las em uma sequência de vídeo em tempo real, permitindo sua utilização tanto para interação como em vídeo digital. Exploramos também algumas técnicas de detecção de características faciais, visando, com estas informações, estimar a orientação da face a partir de pontos na imagem e um modelo de face.

1.3 Contribuições

Como principais contribuições deste trabalho, podemos mencionar:

- desenvolvimento de um sistema de tempo real para detecção e rastreamento de faces baseado em cor e rastreamento de características faciais em correlação com modelos;
- exploração de métodos mais robustos para detecção de características faciais usando Modelos de Aparência Ativa;
- protótipo de um sistema para reconhecimento de faces baseado em Análise de Componentes Principais;
- desenvolvimento de um sistema para indexação automática de vídeo digital, para facilitar a busca de personagens em bibliotecas multimídia;

- desenvolvimento de um sistema para estimação da orientação de face.

O foco deste trabalho foi no estudo e desenvolvimento de técnicas eficientes para detecção e rastreamento de faces e características faciais apropriadas para implementação em tempo real.

Embora nossos resultados possam ser aplicados em interfaces de computadores, os métodos não se mostraram muito precisos. Por isto, exploramos também métodos mais complexos como os Modelos de Aparência Ativa, que por serem computacionalmente mais caros, não podem ser utilizados em tempo real.

Para eliminar falsos candidatos a face antes da fase de rastreamento, utilizamos a técnica de Análise de Componentes Principais. Esta é uma técnica tradicional para redução de dimensionalidade onde os objetos são representados através de seus auto-vetores mais significativos. Esta técnica permite verificar de forma eficiente se uma região candidata possui características de uma face.

Para comprovar a qualidade do nosso sistema, o sistema de detecção de faces foi utilizado para construção automática de índices em vídeo. Um dos principais problemas em bibliotecas multimídia é a busca por conteúdo visual. Por exemplo, atualmente, com a popularização das câmeras digitais, é possível obter um grande volume de fotos pessoais que dificulta a sua organização. Uma aplicação simples de nosso sistema seria selecionar imagens contendo a face de pessoas específicas.

Desenvolvemos também um sistema para estimação da orientação de face utilizando o algoritmo POSIT, que a partir de um conjunto de pontos das características faciais presentes numa imagem de face e de um modelo tridimensional destes pontos, é capaz de estimar a orientação desta face na imagem, permitindo a aplicação deste sistema em interfaces, como por exemplo, reconhecimento de gestos de cabeça, determinação de áreas de interesse, etc.

1.4 Organização da Dissertação

Nos próximos capítulos são apresentados os métodos estudados e implementados e alguns resultados experimentais. O capítulo 2 apresenta uma revisão bibliográfica das técnicas relacionadas a esta pesquisa. O capítulo 3 descreve mais detalhadamente os métodos implementados e a arquitetura do sistema desenvolvido. O capítulo 4 mostra os resultados obtidos na implementação do sistema e dois aplicativos. O primeiro consiste em um aplicativo para indexação de vídeo digital e o segundo consiste de um estimador da orientação de face. O capítulo 5 traz os comentários finais.

Capítulo 2

Revisão Bibliográfica

A pesquisa em visão computacional tem por objetivo executar computacionalmente tarefas atribuídas à visão humana. A visão é o sentido humano mais poderoso e complexo [BH86], que nos permite interagir com o ambiente que nos cerca sem que haja um contato físico direto com este ambiente. A visão computacional se destacou, inicialmente, em aplicações industriais realizadas em ambientes visuais controlados, onde há possibilidade de maior progresso dado as suas restrições. Menor sucesso tiveram as aplicações com objetivo de construir um sistema de visão de propósito geral.

Segundo Horn [BH86], um sistema de visão computacional é um sistema responsável pela análise de imagens para a produção de uma descrição. Esta descrição envolve a captura de aspectos dos objetos presentes na imagem que são úteis para alguma tarefa específica, como por exemplo, interagir com o ambiente.

A área de visão computacional está intimamente relacionada a três áreas: processamento de imagens, reconhecimento de padrões e análise de cenas. A área de processamento de imagens é responsável pela geração de novas imagens a partir de uma imagem de entrada. Ela executa tarefas como eliminação de ruídos e extração de bordas, que são geralmente utilizadas como

pré-processamento em sistemas de visão computacional.

A área de reconhecimento de padrões tem por objetivo classificar um objeto como pertencente a uma determinada classe a partir de características como altura, largura, dentre outras propriedades. Embora não tenha como entrada uma imagem, as técnicas de reconhecimento de padrões são geralmente utilizadas para analisar resultados produzidos por um sistema de visão computacional.

A área de análise de cenas tem a tarefa de transformar uma simples descrição da cena, obtida geralmente de um sistema de visão computacional, em uma descrição mais elaborada que possa ser utilizada por alguma determinada tarefa, como por exemplo, a composição dos objetos presentes na cena e a relação entre eles.

Uma das aplicações de visão computacional mais promissoras é o reconhecimento de faces, que tem recebido atenção especial nos últimos anos motivada pela grande variedade de aplicações comerciais e pelas possibilidades de criação de novas tecnologias com os avanços nesta área de pesquisa [ZCPR03, YKA02]. Um sistema de reconhecimento de face pode ser definido como um sistema que recebe uma imagem ou segmento de vídeo de uma cena, e identifica ou verifica a presença de um ou mais indivíduos nesta cena utilizando um banco de dados de faces de indivíduos. A solução deste problema envolve a detecção de faces nas cenas, extração de características das regiões de face e técnicas de verificação e reconhecimento de face [ZCPR03].

Neste capítulo, apresentamos algumas das principais técnicas para a solução do problema de detecção e rastreamento de faces e características faciais.

2.1 Detecção de faces

Nesta seção, apresentamos uma revisão de algumas das técnicas de detecção de faces estudadas neste trabalho.

Yang *et al.* [YKA02] propuseram uma divisão destas técnicas em quatro categorias, apresentadas a seguir.

2.1.1 Métodos baseados em conhecimento

Nesta categoria, os métodos são desenvolvidos baseados em regras codificadas a partir do conhecimento humano daquilo que se constitui uma face típica. Por exemplo, uma face geralmente aparece com dois olhos que são simétricos um ao outro, uma boca e um nariz. uma possível regra poderia ser criada a partir da relação entre essas características, que pode ser representada pela posição de cada característica e as distâncias relativas entre elas. Primeiro, as características faciais são extraídas e depois as faces são identificadas baseadas nestas regras.

O problema desta abordagem consiste em traduzir o conhecimento humano em regras bem definidas, aplicáveis à maioria das imagens de face. Se tivermos, por exemplo, somente imagens de face frontais, pode ser possível criar regras que apresentem bons resultados na detecção de faces. Porém, fica difícil criar regras detalhadas capazes de abordar todos os possíveis casos quando consideramos as variações da face como variação de orientação, presença de outros componentes como óculos e barba, variação de expressões faciais e oclusão de partes da face.

2.1.2 Métodos baseados em características invariantes

Este métodos procuram extrair características invariantes da face, permitindo a localização destas em condições variadas de orientação e iluminação. Um exemplo são os métodos baseados em detecção de cor de pele. A cor de pele passou a ser um atributo largamente utilizado na

detecção de faces após a observação de que a variedade de cor de pele se dava basicamente pela variação da luminância da cor mesmo considerando pessoas de diferentes etnias.

O RGB é o espaço cromático mais utilizado para representar a cor de um determinado ponto. Porém, para eliminar a informação de luminância da cor, passou-se a utilizar outros espaços cromáticos como o rg-normalizado [YW95] [YLV98] [Vez02] [SAG99], YCrCb [ATAD02] [HAMJ02], L^*u^*v [CM97], entre outros. Terrillon *et al.* [TA00] propôs um novo espaço cromático, o TSL normalizado (*Tint-Saturation-Luminance*), e fez uma análise comparativa de 9 diferentes espaços cromáticos, verificando que o TSL apresenta os melhores resultados para um determinado conjunto de testes, apesar dos resultados sofrerem variação conforme o conjunto de treinamento e o banco de imagens utilizado como teste. Veremos mais detalhes do algoritmo utilizado para segmentação de cor de pele na seção 3.1.

Outro exemplo é o rastreador de pupilas desenvolvido no LaTIn, em colaboração com o Centro de Pesquisas Alcatel da IBM [MF00]. O sistema se utiliza de duas fontes de luz infra-vermelha, uma colocada bem próxima ao centro ótico da câmera (L1) e outra colocada mais distante desse eixo (L2), como vemos na figura 2.1. Cada fonte é formada por um conjunto de 7 LEDs infra-vermelhos, e são ligadas alternadamente. Quando a fonte L1 está ligada, a câmera captura uma imagem onde as pupilas aparecem brilhantes. A fonte L2 gera imagens onde as pupilas aparecem escuras. Esse fenômeno é similar ao que gera pupilas vermelhas em fotografias noturnas tiradas com flash.

Para detectar as faces, primeiramente as imagens de pupilas brilhantes e escuras são subtraídas, resultando em uma imagem onde as regiões de alto contraste são realçadas. A partir dessa imagem diferença, os candidatos a pupilas são filtrados utilizando o conhecimento sobre a geometria esperada das mesmas. Pares de pupilas são agrupados em faces segundo critérios de semelhança e proximidade. A figura 2.2 mostra a aplicação deste método.

Apesar de ser capaz de detectar múltiplas faces, o sistema atual apenas rastreia a face mais

Figura 2.1: Fontes de luz infra-vermelha L1 e L2.

proeminente. Para fins de interação com o usuário, essa face em geral corresponde ao usuário mais próximo ao sistema, e portanto aquele que merece maior atenção.

Embora eficiente, o rastreador de pupilas não pode ser utilizado durante os períodos onde o usuário fecha os olhos. Assim, a face é rastreada até que o rastreador perca as pupilas, ou seja, pare de detectá-las. Para evitar que o rastreador perca a face, um segundo rastreador é utilizado. Esse rastreador utiliza correlação para rastrear apenas uma pequena região ao redor dos olhos. O tamanho pequeno garante um bom desempenho do rastreador por correlação, mas impede que esse rastreador seja utilizado por um período prolongado, devido à acumulação de erros. O rastreador porém se mostra adequado para esse curto período do piscar dos olhos. Mais detalhes sobre o funcionamento do sistema podem ser encontrados em [MF00].

Yow e Cipolla [YC97] propuseram um modelo probabilístico de face baseado em múltiplas aparências da face. Inicialmente, as características faciais são detectadas utilizando um algoritmo de detecção e comparação de bordas. Estas características são então agrupadas e

Figura 2.2: Imagens com as pupilas brilhantes e escuras e a diferença entre as imagens.

comparadas com outros possíveis agrupamentos de características faciais. A partir do modelo probabilístico de face, obtém-se a probabilidade de um dado agrupamento ser uma face ou não e o candidato mais provável é relacionado.

Outro método, proposto por Viola e Jones [VJ01], consiste de um classificador de características em cascata. Estas características são chamadas características do tipo Haar, que são características de bordas, linhas e pontos.

Para construir o classificador, utiliza-se uma modificação do algoritmo de AdaBoost [FS95], escolhendo os melhores filtros de características e organizando-os em estágios (cascata). Geralmente, quanto maior o número de filtros, melhor o resultado da detecção, porém, com maior custo computacional. A construção da cascata, portanto, é uma etapa importante do algoritmo, por ser responsável por eliminar filtros que não trarão benefício à detecção, reduzindo assim o custo computacional.

Cada estágio consiste de um filtro binário onde o resultado do primeiro estágio indica que a característica foi encontrada numa determinada região da imagem. Seguem para o estágio seguinte somente as regiões que passaram pelos estágios anteriores, até chegar ao final do processo.

2.1.3 Métodos baseados em ajustes de modelos de forma

Nesta categoria, temos um modelo de face padrão, que pode ser definido manualmente ou parametrizado por uma função. Este modelo é composto pelo contorno da face e suas características, podendo ser fixos ou deformáveis. Neste último caso, ele é definido por diversos exemplos que captam as possíveis variações de forma da face. Dada uma imagem de entrada contendo uma face, o modelo de forma da face se ajusta à imagem da face, determinando a sua localização.

Estes métodos não se aplicam propriamente a detecção de faces, mas assumem uma detecção aproximada da face para então ajustar o modelo à imagem. O que eles fazem é fornecer uma descrição mais precisa e detalhada da face, sendo úteis para outros algoritmos de reconstrução, extração de expressões faciais e estimação da orientação de face.

Um exemplo desta abordagem são os modelos de forma ativa (*Active Shape Models*) proposto por Cootes e Taylor [CT01]. A seção 3.4 apresenta mais detalhes desse algoritmo.

2.1.4 Métodos baseados em aparência

Diferente dos modelos de forma, os métodos baseados em aparência captam a variação da aparência da face aprendida a partir de um conjunto de imagens de face. PCA (*Principal Component Analysis*) ou Análise de Componentes Principais, proposto por Turk e Pentland [TP91], é um exemplo de um método baseado em aparência. A partir do conjunto de treinamento, é criado um espaço de faces formado por vetores chamados *eigenfaces*, e as imagens de face podem ser descritas por uma combinação linear destas *eigenfaces*. Quando a projeção de uma nova imagem neste espaço de faces se aproxima da projeção de uma face, identificamos esta imagem como contendo uma face. A seção 3.2 apresenta mais detalhes desse método.

2.2 Rastreamento de faces

O rastreamento de faces consiste na estimação do movimento das faces em uma seqüência de vídeo. Podemos rastrear a face como um todo, como se tratando de um objeto rígido, ou rastrear características faciais, considerando as deformações que a face pode sofrer.

Temos métodos baseados no cálculo do fluxo ótico, que é uma medida do movimento aparente da luminosidade da imagem, assumindo que a luminosidade de pontos dinâmicos ou estáticos de um objeto não se altera no decorrer do tempo. Um exemplo desses métodos é a técnica de Lucas & Kanade [LK81].

Outra técnica de rastreamento consiste em determinar um modelo estocástico de movimento da face para estimar o seu estado futuro. O filtro de Kalman é um exemplo de estimador que tem sido largamente utilizado em rastreamento de faces e características faciais [WB95]. Ele é composto de duas fases, uma de previsão e outra de correção, que se repetem ciclicamente até obtermos convergência entre o estado previsto e o observado. Na fase de previsão utilizamos o histórico de estados e um modelo de comportamento para determinar o próximo estado do sistema. Na fase de correção, fazemos uma observação do sistema e verificamos o quanto a previsão se aproximou do estado observado. Esta informação é utilizada para melhorar o modelo de comportamento e atualizar o histórico de estado, para então voltarmos à fase de previsão.

O CAMShift [CM97] é um método bastante eficiente de rastreamento, que considera o histograma do objeto rastreado e a posição deste no quadro anterior para estimar o movimento deste objeto, em nosso caso, a face.

No próximo capítulo, veremos com mais detalhes os métodos aplicados em nosso sistema de detecção e rastreamento de faces e características faciais.

Capítulo 3

Detecção e Rastreamento de Faces e Características Faciais

Neste capítulo, apresentamos uma descrição teórica dos diversos métodos utilizados em nosso sistema de detecção e rastreamento de faces e características faciais.

Dada uma imagem de entrada, primeiramente, detectamos as regiões de cor de pele e identificamos as regiões candidatas a conter uma face. Utilizando o método do gradiente, detectamos os candidatos a olhos, que são então pareados baseados em algumas regras heurísticas, e cada par define então uma região candidata a face. Utilizando PCA, escolhemos qual o candidato que mais se aproxima de uma face. Uma vez detectada a face, localizamos as demais características faciais por correlação. Para evitar resultados inconsistentes com uma face real, utilizamos modelos deformáveis para determinar se as características localizadas estão de acordo com um modelo de face.

Como saída do programa, temos a face e as características faciais detectadas na imagem. A figura 3.1 mostra um diagrama de como estes métodos foram utilizados em nosso sistema.

Figura 3.1: Diagrama do sistema de detecção de faces e características faciais.

3.1 Detecção de Cor de Pele

A cor de pele tem sido a característica mais utilizada para rastreamento devido a sua simplicidade e eficiência computacional. Esta técnica consiste em obter a distribuição de cor de pele em um espaço cromático determinado a partir de um conjunto de treinamento formado por amostras de cor de pele. Entre os espaços mais utilizados estão o rg-normalizado [YW95] [YLW98] [Vez02] [SAG99], YCrCb [ATAD02] [HAMJ02], L^*u^*v [CM97], entre outros.

No espaço de cores RGB tradicional cada cor é identificada pela intensidade de vermelho,

verde e azul. O espaço de cores rg-normalizado é um espaço de cores bidimensional que não possui a informação de luminância. Neste espaço, uma cor é representada pela proporção de vermelho, verde e azul ao invés da intensidade absoluta de cada componente. Uma vez que a soma dessas proporções é sempre igual a 1, podemos considerar apenas a proporção de vermelho e verde e descartar a proporção de azul (pois esta é função das outras duas) e conseqüentemente uma cor neste espaço pode ser representada pela dupla (r, g) .

Apesar de uma dupla (r, g) possuir menos informação do que uma equivalente tripla (R, G, B) , a primeira é normalmente útil nas aplicações em que queremos justamente considerar apenas a informação de crominância e ignorar a luminância.

Terrillon *et al.* [TA00] propôs um novo espaço cromático chamado TSL. Neste espaço cada cor é representada por uma tripla (T, S, L) onde cada componente tem o seguinte significado: Matiz (T - Tint): informação da cor; Saturação (S): diluição da cor original com a luz branca; e Luminância (L): quantidade de energia de uma fonte de luz percebida por um observador.

Dada uma tripla (R, G, B) representando a intensidade de vermelho, verde e azul, podemos determinar a equivalente tripla (T, S, L) representando a matiz, a saturação e a luminância através da seguinte relação:

$$S = [9/5(r'^2 + g'^2)]^{1/2} \quad (3.1)$$

$$T = \begin{cases} \arctan(r'/g')/2\pi + 1/4, & \text{se } g' > 0 \\ \arctan(r'/g')/2\pi + 3/4, & \text{se } g' < 0 \\ 0 & \text{se } g' = 0 \end{cases} \quad (3.2)$$

$$L = 0.299R + 0.587G + 0.114B \quad (3.3)$$

onde $r' = (r - 1/3)$, $g' = (g - 1/3)$, $r = R/(R + G + B)$ e $g = G/(R + G + B)$.

A distribuição de cor de pele para este modelo de cor foi obtida utilizando uma normalização do TSL pelas componentes T e S, o chamado TS-normalizado.

Em Pera *et al.* [PBM04], foi feita uma análise da distribuição de cor de pele neste espaço e verificou-se que a componente T possuía uma variância significativamente menor que a componente S. Isto nos motivou a utilizar somente a componente T. A utilização desta componente foi capaz de gerar uma boa segmentação de cor de pele se comparado aos outros espaços citados anteriormente, embora apresente grande quantidade de falsos positivos em regiões da imagem de tonalidade vermelha.

Fase de Treinamento

A primeira etapa deste algoritmo consiste na fase de treinamento. Esta fase é responsável por descrever a distribuição das cores no espaço cromático determinado.

A partir de imagens coloridas de face, extraímos manualmente regiões de cor de pele para formar o banco de imagens utilizadas no treinamento. A figura 3.2 exemplifica a obtenção de uma imagem do conjunto de treinamento.

Figura 3.2: Extração de regiões de cor de pele para conjunto de treinamento.

Supondo que a distribuição de cor de pele pode ser representada por um modelo de distribuição gaussiana, tal distribuição pode ser determinada pela média m e pela matriz de covariância C das amostras x de cores de pele.

Fase de Classificação

A distribuição da cor de pele obtida na fase de treinamento é utilizada para atribuir uma probabilidade para cada pixel da imagem conforme a sua cor. Esta probabilidade é calculada através da distância de Mahalanobis, utilizando a média e a covariância calculadas na fase de treinamento.

A probabilidade é dada por:

$$P(x) = \exp[-0.5(x - m)^T C^{-1}(x - m)] \quad (3.4)$$

Este cálculo resulta em uma nova imagem em níveis de cinza indicando a probabilidade do pixel correspondente ser cor de pele, como vemos na figura 3.3. Nesta, as regiões mais escuras correspondem à maior probabilidade de cor de pele.

Figura 3.3: Resultado do cálculo da probabilidade de cor de pele para cada pixel.

O próximo passo do algoritmo consiste em um processo de limiarização. O problema neste passo é definir um limiar aplicável a diferentes imagens. A variação de cor de pele dificulta a

escolha de um limiar fixo para limiarizar a imagem. A solução é aplicar um limiar adaptativo.

Analisando o histograma da imagem de probabilidade, temos as regiões de cor de pele representadas pelos maiores valores de probabilidade. Mas geralmente temos várias outras regiões da imagem que apresentam valores intermediários de probabilidade. O objetivo é encontrar um limiar ótimo que separe essas duas regiões.

Primeiramente, precisamos definir qual a faixa aceitável a conter o limiar. Após alguns testes verificamos que um limiar no intervalo entre 50% e 90% seria aceitável para diferentes cores de pele.

Um limiar bom é aquele que inclui o maior número de pontos da face e exclui pontos que não pertencem a face. Supondo que a face encontra-se na região superior do histograma e que outros objetos encontram-se em regiões inferiores do histograma, o ponto que separa a região da face é caracterizado por estar contido em um intervalo do histograma com o menor número de pontos.

Com base nesta observação, o processo de limiarização adaptativa consiste em particionar o histograma de probabilidade em pequenos intervalos, e encontrar o intervalo que possui o menor número de pontos dentre aqueles que são candidatos a conter um limiar adequado. Escolhemos então o limiar dentro deste intervalo.

Figura 3.4: Exemplo de histograma particionado em intervalos de 10/100.

A figura 3.4 mostra um exemplo de um histograma particionado em intervalos de 10%.

Para este histograma, considerando somente o intervalo entre 50% e 90%, escolhemos o limiar dentro do intervalo de menor área, no caso, entre 51% e 60%. A figura 3.5 mostra o resultado da aplicação deste limiar para a figura 3.3.

Figura 3.5: Resultado da limiarização aplicada ao exemplo dado na figura 3.3.

O processo de limiarização resulta numa imagem binária onde cada pixel é classificado como cor de pele ou não-pele. Para eliminar ruídos desta imagem, aplicamos a operação morfológica de abertura (erosão seguida por dilatação) e consideramos como regiões candidatas a cor de pele somente as componentes conexas cuja área seja superior a uma área mínima determinada de acordo com o tamanho mínimo esperado da face. A figura 3.6 mostra o resultado do processo de limiarização seguida de eliminação de ruído.

3.2 Análise de Componentes Principais

O objetivo da detecção de cor de pele é de gerar regiões candidatas a faces. Nesta seção, veremos um método que, dada uma região da imagem, responde com maior precisão se esta corresponde a uma região de face ou não. Este método, proposto por Turk e Pentland [TP94],

Figura 3.6: Resultado da limiarização seguida de operações para eliminação de ruídos.

é conhecido como Análise de Componentes Principais, PCA (*Principal Components Analysis*) ou transformada de Karhunen-Loève. Este método tem sido amplamente utilizado para verificação e reconhecimento de faces, e é a base do método de interpretação dos modelos de forma de objetos discutido na seção 3.4, os Modelos Deformáveis.

Diferentemente da maioria das abordagens, em PCA, as características da face não estão, necessariamente, relacionadas à noção intuitiva de aspectos da face, como olhos, nariz, lábios e cabelos. As características codificadas correspondem à variação das faces obtidas a partir de uma coleção de imagens de face.

A partir desta coleção, obtemos o que chamamos de *eigenfaces*, que são as componentes principais deste espaço de faces. Cada imagem deste conjunto pode ser descrito por uma combinação linear das *eigenfaces*.

Em termos matemáticos, as *eigenfaces* correspondem aos auto-vetores da matriz de covariância de um conjunto de imagens, tratando cada imagem como um ponto (ou vetor) de dimensão muito grande.

A classificação de imagens por PCA consiste dos seguintes passos:

- A partir de um conjunto de treinamento, calculamos as *eigenfaces*, ou seja, os auto-

vetores da matriz de covariância das imagens do conjunto, e escolhemos dentre elas, aquelas que apresentam maior variância dos padrões. Estas correspondem às que possuem maior auto-valor associado. As imagens escolhidas definem o **espaço de faces**.

- Dada uma imagem candidata à face, projetamos esta imagem no **espaço de faces** e verificamos se esta projeção se aproxima da projeção de uma face neste espaço. Com isso, saberemos se esta imagem corresponde ou não à imagem de uma face.

Cálculo da Matriz de Covariância

Seja uma imagem de face de tamanho $N \times N$, construímos um vetor Γ_i correspondente no espaço (N^2) -dimensional, dispondo sucessivamente as colunas da imagem.

A partir do conjunto de treinamento de M imagens, teremos: $\Gamma = [\Gamma_1, \Gamma_2, \dots, \Gamma_M]$. Seja a face média definida por:

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n$$

Cada face difere da média pelo vetor:

$$\Phi_i = \Gamma_i - \Psi$$

A partir destes vetores, podemos calcular a matriz de covariância C , que será dada por:

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T \quad (3.5)$$

ou

$$C = AA^T, \text{ onde } A = [\Phi_1, \Phi_2, \dots, \Phi_M] \quad (3.6)$$

Como podemos ver, a matriz C é uma matriz $N^2 \times N^2$, cujos elementos da diagonal $C_{p,p}$ corres-

pondem à variância da característica p , e os demais elementos $C_{p,q}$ correspondem à covariância entre as características p e q . Dado que C é real e simétrica, é sempre possível encontrar um conjunto de N^2 auto-vetores ortonormais. Este número, porém, é computacionalmente caro para os tamanhos típicos de imagem, como por exemplo 256×256 .

Ao mesmo tempo, queremos encontrar somente os auto-vetores que apontem para as direções de maior variância dos padrões. Estes são os que possuem maiores auto-valores associados.

Se o total de pontos M no **espaço de face**, ou seja, o total de imagens, é menor que a dimensão do espaço ($M < N^2$), teremos apenas $M - 1$ auto-vetores significativos, sendo que os demais terão auto-valores associados nulos.

Felizmente, é possível obter os auto-vetores significativos desta matriz a partir de uma outra matriz $M \times M$. Considere os auto-vetores v_i de $A^T A$:

$$A^T A v_i = \lambda v_i$$

Multiplicando à esquerda ambos os lados por A , temos:

$$A A^T A v_i = \lambda A v_i \quad (3.7)$$

Assim, $A v_i$ são os autovetores significativos de $C = A A^T$.

Para encontrar as *eigenfaces* significativas de um conjunto de M faces, construímos a matriz $L = A^T A$ e achamos os auto-vetores $v_l, l = 1, \dots, M$ dessa matriz. A partir destes, obtemos os autovetores u_l de C da seguinte maneira:

$$u_l = \sum_{k=1}^M v_{l_k} \Phi_k, \quad l = 1, \dots, M \quad (3.8)$$

Os auto-vetores u_i de C são chamados de *eigenfaces*, devido à aparência de face que tomam ao serem reescritos na forma de imagem.

Classificação de Imagens

Temos, então, as *eigenfaces* que descrevem o espaço de faces. Dada uma nova imagem Γ , podemos classificá-la como pertencente ao espaço de faces ou não. Para isto, o primeiro passo consiste em projetar esta imagem no espaço, o que é feito pela seguinte operação:

$$\omega_k = u_k^T(\Gamma - \Psi), \quad k = 1, \dots, M \quad (3.9)$$

Os pesos ω_k formam o vetor $\Omega^T = [\omega_1, \omega_2, \dots, \omega_M]$ que descreve a contribuição de cada *eigenface* u_k na representação da imagem.

A partir desta projeção, podemos calcular a distância euclidiana entre a imagem Γ ajustada pela média das faces, $\Phi = \Gamma - \Psi$, e a projeção desta imagem no espaço de faces, $\Phi_f = \sum_{i=1}^M \omega_i u_i$. A distância euclidiana é dada por:

$$\epsilon^2 = \|(\Phi - \Phi_f)\|^2 \quad (3.10)$$

A imagem será classificada como imagem de face se o valor de ϵ for menor que um limiar estabelecido experimentalmente, θ_ϵ .

Implementação

Um dos problemas apresentados pela técnica PCA está no fato deste não ser capaz de distinguir entre a face e o fundo da imagem. Turk e Pentland [1991] propuseram multiplicar a imagem por uma gaussiana bidimensional centrada na face, diminuindo o fundo e acentuando o centro da face. A nossa solução foi utilizar uma região mais fechada da imagem, que não contém

Figura 3.7: Imagens de treinamento de PCA (ao alto) e face média.

o fundo, mas somente a parte interna da face, compreendendo somente os olhos, nariz e boca. Dessa forma, temos um espaço de faces mais uniforme, que além de não conter o fundo, também não contém o cabelo, ouvidos e queixo, que são regiões da face de maior variabilidade. A figura 3.7 nos mostra alguns exemplos de imagens utilizadas no treinamento e a face média do treinamento.

Outro problema está na diferença de escala da imagem de entrada com relação às *eigenfaces*. A solução que adotamos neste caso foi alinhar a imagem de entrada às *eigenfaces* a partir da detecção de pontos correspondentes.

Para alinhar a imagem de entrada às *eigenfaces*, utilizamos como ponto de controle o centro da íris dos olhos. Na seção 3.3 veremos técnicas para detecção destes pontos. O alinhamento é feito através de uma transformação afim que leva os pontos de controle das *eigenfaces* aos pontos correspondentes da imagem de entrada.

3.3 Detecção de Características Faciais

A extração de características faciais é um passo fundamental para a estimação da orientação de faces. As características faciais compreendem todos os elementos presentes e que determinam

uma face: pele, boca, nariz, narinas, olhos, sobrancelhas, queixo, etc.

Algumas destas características são mais facilmente extraídas da imagem. Na seção 3.1 vimos métodos de detecção de faces a partir da detecção de cor de pele. Esta característica é facilmente extraída sob certas condições, como na presença de uma distribuição adequada que forneça uma boa descrição da cor de pele.

Quando escolhemos as características faciais a serem rastreadas, consideramos a facilidade de detecção desta característica e a sua utilidade na construção de um modelo de face que possibilite a estimação da orientação da face da forma mais precisa possível.

As diversas técnicas de detecção de características faciais podem ser classificadas como:

- técnicas baseadas em luminância e cromaticidade [Vez02, MF00, YS02];
- técnicas baseadas em geometria e simetria facial [RY92, GH01];
- técnicas baseadas em *template matching* [YC97, ETC98, KM96, VJ01].

As técnicas baseadas em *template matching*, embora apresentem resultados satisfatórios, são sensíveis a variações de escala da face, aumentando a complexidade computacional do algoritmo. Já as técnicas baseadas em luminância e cromaticidade e as baseadas em geometria e simetria facial, geralmente precisam ser aplicadas em conjunto com outros métodos para se obter um resultado satisfatório.

Uma limitação importante da maioria das técnicas apresentadas é o fato delas somente detectarem características de faces frontais. Como nosso objetivo é que estas características sirvam para estimação da orientação da face, o domínio de nossa aplicação consiste tanto de imagens de faces frontais quanto de faces com alguma rotação. Com isto, pode ocorrer de, dependendo da orientação da face, não termos a imagem de algumas características faciais que estavam sendo rastreadas, como, por exemplo, um dos olhos, sobrancelhas e narinas.

Dedicamos maior atenção às técnicas de detecção de olhos em imagens coloridas. Na próxima seção descrevemos as técnicas utilizadas em nosso trabalho.

3.3.1 Detecção de olhos pelo gradiente

O olho humano é a característica facial mais utilizada para detecção de faces. A detecção dos olhos fornece informações importantes que facilitam a detecção de outras características faciais.

Primeiramente, dada a simetria da face, a distância entre os olhos nos fornece uma estimativa do tamanho da face, ou seja, a escala. O erro desta estimativa ocorre basicamente devido à sua orientação tridimensional.

Outra importante informação é a orientação bidimensional da face que é dada pelo ângulo entre a reta que passa pelos olhos e o eixo horizontal da imagem. Esta informação permite estimarmos uma posição inicial de busca de outras características faciais.

O próprio olho humano já é uma grande fonte de informação para diversos sistemas de interação. Em nosso laboratório, tivemos pesquisas recentes na área de interfaces prestativas [Bar05], em que a direção do olho fornecia a informação do foco de atenção do usuário com relação à tela de um monitor. Esta informação permitiu o desenvolvimento de um sistema em que o olho é utilizado como dispositivo de apontamento, auxiliando ou mesmo substituindo o mouse.

Para detectar olhos, nos concentramos em duas características básicas do olho humano: o formato aproximadamente circular da íris e o contraste entre a íris (parte colorida) e a esclera (parte branca do olho).

O primeiro passo dos algoritmos que estudamos consiste na aplicação de métodos de detecção de bordas. Para isso, utilizamos filtros gradiente.

Filtros gradiente

Seja a imagem uma função $f : (x, y) \rightarrow \mathbb{R}$. As derivadas de primeira ordem de f nos fornecem uma medida de transição dos níveis de cinza entre pontos vizinhos. Quando temos uma região de transição entre pontos claros e escuros, a 1ª derivada assume valores positivos, e numa região com níveis de cinza constante, assume valor zero. Além disso, quanto maior a transição dos níveis de cinza entre pontos claros e escuros, maior a magnitude das 1ªs derivadas [GW02].

A partir desta observação, podemos concluir que a magnitude das 1ªs derivadas pode ser utilizada para detectar a presença de bordas nos pontos da imagem.

As derivadas de primeira ordem de uma imagem são dadas pelo vetor gradiente. O gradiente de uma imagem $f(x, y)$ no ponto (x, y) é definido por:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}. \quad (3.11)$$

onde:

$$\nabla f = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{\frac{1}{2}}. \quad (3.12)$$

A partir destas equações, notamos que o vetor gradiente aponta para a direção de maior mudança dos níveis de cinza do ponto (x, y) , e que a magnitude nos fornece uma medida do crescimento de $f(x, y)$ na direção do gradiente.

Podemos definir um ponto numa imagem como sendo ponto de borda se a magnitude de sua derivada bidimensional de primeira ordem é superior a um limiar pré-estabelecido.

O cálculo do gradiente de uma imagem envolve obter as derivadas parciais para cada ponto desta imagem. No caso discreto, podemos aplicar filtros lineares. Seja uma imagem $f(x, y)$ de tamanho $M \times N$ e uma função $w(x, y)$ de tamanho $n \times n$ que chamamos de máscara, o filtro

da imagem f com a máscara w é dado por:

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t), \quad (3.13)$$

onde $a = (m - 1)/2$, $b = (n - 1)/2$, $x = 0, \dots, M - 1$ e $y = 0, \dots, N - 1$.

Seja, por exemplo, w uma máscara de tamanho 3×3 . Computacionalmente, o processo de filtragem equivale a aplicar esta máscara a todos os pontos da imagem, somando o produto de cada par de células correspondentes. Este processo é chamado de convolução.

$w(-1, -1)$	$w(-1, 0)$	$w(-1, 1)$
$w(0, -1)$	$w(0, 0)$	$w(0, 1)$
$w(1, -1)$	$w(1, 0)$	$w(1, 1)$

Máscara

$f(x - 1, y - 1)$	$f(x - 1, y)$	$f(x - 1, y + 1)$
$f(x, y - 1)$	$f(x, y)$	$f(x, y + 1)$
$f(x + 1, y - 1)$	$f(x + 1, y)$	$f(x + 1, y + 1)$

Janela da imagem

Filtro de Sobel

Entre as máscaras mais utilizadas para o cálculo do gradiente está a máscara de Sobel, que é definida por:

$$G_x = \begin{array}{|c|c|c|} \hline -1 & 0 & +1 \\ \hline -2 & 0 & +2 \\ \hline -1 & 0 & +1 \\ \hline \end{array} \quad (3.14)$$

e

$$G_y = \begin{array}{|c|c|c|} \hline +1 & +2 & +1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array} \quad (3.15)$$

A figura 3.8 mostra o resultado da aplicação do filtro de Sobel na imagem da figura 3.3. Como dissemos anteriormente, para determinar os pontos de borda, podemos escolher um limiar adequado e determinar que todos os pontos acima deste limiar são pontos de borda. A figura 3.9 mostra o resultado para alguns valores de limiares.

Como vemos nestas figuras, a simples escolha de um limiar não se mostra adequado para obtermos um contorno mais fino que nos permitisse detectar a região circular dos olhos. Vejamos, então, uma nova solução que nos oferece uma melhor solução para detecção de contornos, o algoritmo de detecção de borda de Canny [Can86].

Algoritmo de Canny para detecção de bordas

O algoritmo de Canny para detecção de bordas [Can86] é tido por muitos como um detector de bordas ótimo. Ele determinou alguns critérios com o objetivo de melhorar o desempenho dos métodos de detecção de bordas. Os critérios são os seguintes:

Figura 3.8: Resultado da aplicação do filtro de Sobel na direção x (à esquerda), y (à direita) e a magnitude (abaixo).

1. Baixa taxa de erro: isto significava detectar todas as bordas presentes na imagem e, ao mesmo tempo, evitar os resultados falso-positivos;
2. Pontos de borda bem localizados: ou seja, o algoritmo deveria minimizar a distância entre a borda detectada e a borda real da imagem;
3. Somente uma resposta para cada borda: isto significava que cada borda deveria ser representada somente por uma linha, ou seja, cada ponto de borda poderia ter no máximo 2 vizinhos que também são pontos de borda, diferente dos resultados da limiarização do filtro de Sobel, em que alguns pontos de borda tinham todos os seus vizinhos também como pontos de borda.

Figura 3.9: Resultado da limiarização sobre o resultado do filtro de Sobel com limiares de 20 (à esquerda), 30(à direita) e 40 (abaixo).

Os primeiros passos do algoritmo de Canny consistem na aplicação do filtro de Sobel para encontrar o vetor gradiente de cada ponto da imagem, bem como sua magnitude e direção.

O passo seguinte consiste em relacionar a direção do vetor gradiente com uma direção que pode ser traçada na imagem. Seja um pixel p :

$$G_x = \begin{array}{|c|c|c|} \hline 3 & 2 & 1 \\ \hline 4 & p & 0 \\ \hline 5 & 6 & 7 \\ \hline \end{array}$$

Temos somente 4 direções possíveis que podem ser traçadas na imagem: 0° na direção dos pixels 0 e 4, 45° na direção dos pixels 1 e 5, 90° na direção dos pixels 2 e 6 e 135° na direção de 3 e 7. Assim, dada a direção do gradiente de um ponto, esta é setada para o valor mais

próximo entre as 4 direções possíveis de serem traçadas.

Feito isto, o próximo passo consiste em aplicar a supressão de não máximos. Isto é feito tomando um ponto de borda e percorrendo o restante da borda ajustando para zero aqueles pontos que não são considerados borda. Com isto teremos uma linha fina na imagem de saída.

Finalmente, uma histeresse é utilizada para evitar quebras de borda. Para isto são definidos dois valores de limiar T_1 e T_2 , $T_1 > T_2$. Primeiramente, marcamos todos os pontos maiores que T_1 como sendo pontos de borda. Feito isto, todos os pontos conexos a estes pontos de borda com valores maiores que T_2 também são considerados pontos de borda. Dessa forma, para se obter uma borda, iniciamos com um ponto cuja magnitude do gradiente é maior que T_1 , e a borda só termina quando não tivermos mais gradientes com magnitude superior a T_2 .

Figura 3.10: Resultado do algoritmo de Canny para extração de bordas.

A figura 3.10 mostra o resultado da aplicação do método de Canny na imagem de face da figura 3.3. Como podemos ver, temos agora o contorno da íris aproximadamente circular. Logo, podemos aplicar, sobre essa imagem, algoritmos para detecção de contornos circulares. Um algoritmo conhecido é a transformada de Hough para círculos. O segundo, proposto por Kothari e Mitchel [KM96] utiliza a direção do gradiente para criar um algoritmo capaz de detectar o centro da íris.

Apresentamos, agora, alguns algoritmos que utilizamos para detecção de olhos, através das imagens obtidas pela aplicação dos filtros de gradiente.

Transformada de Hough para detecção de círculos

A transformada de Hough é muito utilizada e conhecida como um algoritmo para detecção de linhas retas, porém ele pode ser também aplicado para a detecção de círculos.

Assumindo que temos uma imagem binária, que pode ser obtida pelos filtros de gradiente vistos anteriormente, queremos detectar linhas retas ou curvas na imagem. Logicamente, queremos detectar somente as maiores linhas, ou seja, aquelas formadas por um maior número de pontos. Vejamos inicialmente o algoritmo para detecção de linhas retas.

Seja um ponto (x_i, y_i) da imagem e a equação geral da reta dada por:

$$y_i = ax_i + b. \quad (3.16)$$

Como sabemos, existem infinitas linhas que passam pelo ponto (x_i, y_i) , cada uma com diferentes valores de a e b . No entanto, se reescrevermos a equação 3.16:

$$b = -ax_i + y_i, \quad (3.17)$$

e considerarmos o plano ab , que é o espaço de parâmetros, teremos a equação de uma única linha associada ao ponto (x_i, y_i) . Um segundo ponto (x_j, y_j) também terá uma linha associada no espaço de parâmetros.

As retas associadas aos pontos (x_i, y_i) e (x_j, y_j) irão se interceptar no plano ab no ponto (a', b') , onde a' representa a inclinação e b' o ponto de intersecção com o eixo y da linha que contém (x_i, y_i) e (x_j, y_j) no plano xy . Ou seja, para uma linha no plano xy , os pontos desta linha terão retas associadas no plano ab que irão se interceptar no mesmo ponto (a', b') no

espaço de parâmetros. Precisamos, portanto, de um algoritmo que detecte estes pontos de intersecção. Isto é feito utilizando acumulação.

Seja uma matriz de acumulação A cujas células representam a quantização dos valores de a entre a_{min} e a_{max} e de b entre b_{min} e b_{max} . A matriz é inicialmente zerada. Para cada ponto de linha (x_k, y_k) no plano xy , varremos os valores de a entre a_{min} e a_{max} e obtemos o valor de b correspondente através da equação 3.17. Se necessário, arredondamos o valor de b para o valor quantizado mais próximo. Se a escolha de um a_p resulta em uma solução b_q dentro dos respectivos domínios, incrementamos a célula $A(p, q)$ correspondente da matriz de acumulação.

No final, teremos a matriz A indicando, em cada célula $A(i, j)$, o total de pontos colineares na reta $y = a_i x + b_j$. Estabelecemos, então, um limiar para detectar as linhas presentes na imagem.

Existem outras formas de utilização deste algoritmo para resolver o problema de não termos uma limitação para os valores de a e b . Para mais detalhes, veja [DH72].

Podemos dizer que a transformada de Hough consiste em mapear cada pixel da imagem para o espaço de parâmetros, que é organizado em forma de um acumulador n -dimensional, onde n corresponde ao número de parâmetros.

A partir desta definição, a aplicação da transformada de Hough para a detecção de círculos consiste apenas em redefinir o espaço de parâmetros, que neste caso será tridimensional, indexado por três parâmetros que especificam a localização de um círculo de centro $a = (a_1, a_2)$ e o raio r . Para isto, utilizamos a representação paramétrica do círculo:

$$(x - a_1)^2 + (y - a_2)^2 = r^2. \quad (3.18)$$

No final, teremos a matriz de acumulação indicando, em cada célula $A(a_i, a_j, r_k)$, o total de pontos cujos centros são dados por (a_i, a_j) e raio r_k . Estabelecemos, então, um limiar para

o caso da íris para detectar as linhas curvas presentes na imagem.

A figura 3.11 mostra a imagem representando a matriz de acumulação e os pontos escolhidos como candidatos a olhos.

Figura 3.11: (a) Matriz de acumulação para raios entre 6 e 9, (b) Pontos candidatos a olhos em azul e par de olhos escolhido pelo fator J (vide seção 3.3.1) em amarelo.

Detecção dos olhos pela direção do gradiente

Kothari e Mitchel [KM96] propuseram um algoritmo para detecção do centro da íris baseado em algumas observações com relação à direção do gradiente.

Os vetores gradiente apontam sempre para a região de vizinhança mais clara do ponto. Na região da íris, teremos os gradientes apontando para a direção oposta ao centro da íris. Isto sugere um novo algoritmo de acumulação, se invertermos o sentido destes vetores.

O algoritmo consiste dos seguintes passos:

1. Calculamos os vetores gradiente de cada ponto;
2. Tomamos uma matriz de acumulação de mesmo tamanho da imagem, e a iniciamos com zero;

3. Nos pontos de borda, obtemos um segmento de reta partindo do ponto, no sentido contrário do gradiente. Nos pontos pertencentes a este segmento, incrementamos os pontos correspondentes da imagem de acumulação.

Como resultado deste algoritmo, teremos uma imagem cujos pontos de valor máximo são potenciais candidatos a centro de íris.

Os vetores gradiente podem ser calculados utilizando o filtro de Sobel na direção x e y . Para determinar sobre quais pontos iremos aplicar o passo 3 do algoritmo, podemos aplicar um limiar mínimo sobre a magnitude do gradiente de cada ponto. Ainda no passo 3, o tamanho do segmento é uma variável importante do sistema. Um valor razoável seria o valor máximo de raio esperado da íris. Para obter este segmento, utilizamos o algoritmo de Bresenham [Enc3].

A figura 3.12 mostra a imagem representando a matriz de acumulação e os pontos escolhidos como candidatos a olhos.

Figura 3.12: (a) Matriz de acumulação criada pelo algoritmo de detecção pela direção do gradiente, (b) Pontos candidatos a olhos em azul e par de olhos escolhido pelo fator J em amarelo.

Tanto na transformada de Hough quanto no algoritmo de detecção pela direção do gradiente, temos como resultado diversos pontos candidatos a olhos. Isto ocorre devido a ruídos na imagem e a outras regiões semi-circulares de face que são captados pelos algoritmos, gerando

resultados falso-positivos. A próxima tarefa é, portanto, escolher o par de pontos correspondente aos olhos.

Escolhemos somente os pontos que estão dentro de uma região de cor de pele. A distância entre o par de pontos deve estar dentro de um intervalo relacionado ao tamanho da região de cor de pele. Além disso, supondo que os pontos dos olhos estão a uma altura de $2/3$ da altura da região de cor de pele e que ambos os pontos estão aproximadamente alinhadas na horizontal, calculamos um fator J para cada um dos pares (x_i, y_i) e (x_j, y_j) da seguinte maneira:

$$J_{i,j} = |y_i - y_j| + \text{dist}(m, p), \quad (3.19)$$

onde m é o ponto médio entre (x_i, y_i) e (x_j, y_j) e, seja a região de pele uma janela de tamanho $W \times H$, $p = (w/2, h/3)$ no sistema de coordenadas desta janela (cuja origem é localizada no canto superior esquerdo).

Escolhemos, então, os pares com menor fator J e aplicamos PCA utilizando estes pontos como referência para alinhamento com as eigenfaces. O par de pontos que resultar em uma menor norma da aplicação de PCA é considerado o par de pontos correspondentes aos olhos, onde cada ponto representa o centro da íris.

3.3.2 Detecção de características por correlação

Na seção anterior, vimos formas de detecção dos olhos. Para detectar outras características como narinas e cantos da boca utilizamos correlação.

A correlação é uma operação bastante parecida com a convolução. Enquanto a convolução é utilizada para filtrar imagens, como por exemplo, os filtros gradiente, a correlação é uma operação que nos permite fazer uma comparação entre imagens.

Figura 3.13: Exemplos das características faciais utilizadas para cálculo da correlação.

Sejam duas imagens dadas por duas funções $f(x, y)$ e $h(x, y)$ de tamanho $M \times N$. A correlação dessas duas funções é definida por:

$$f(x, y) \circ h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(x, y) h(x + m, y + n), \quad (3.20)$$

onde $f^*(x, y)$ corresponde ao conjugado de $f(x, y)$. Como estamos lidando com funções reais, temos que $f^*(x, y) = f(x, y)$.

O resultado da correlação é um número que indica a semelhança entre as funções f e h . Quando estas funções forem iguais, a correlação terá o maior valor possível.

A detecção de características por correlação consiste em primeiramente obtermos uma amostra da característica que queremos detectar. Esta amostra, que corresponde à função $h(x, y)$, foi obtida a partir de média de diversos exemplos da característica extraídos do banco de imagens de faces, como mostra a figura 3.13. Estes exemplos foram obtidos através da marcação manual de pontos de características faciais, que foram devidamente alinhados e escalonados para sua extração.

Figura 3.14: Exemplo de aplicação de correlação. A primeira linha mostra a janela $f(x, y)$, a segunda a amostra $h(x, y)$, a terceira o resultado da correlação e a quarta mostra a posição da característica.

Seja $f(x, y)$ uma janela na imagem que contém a característica, aplicamos a correlação entre as funções f e h . Fazemos isto por toda região de interesse da imagem, resultando numa nova imagem cujos máximos indicarão a maior probabilidade de ocorrência da característica.

Um dos problemas do uso desta abordagem para detecção de características é a escala. Em nosso sistema, isto foi resolvido pelo fato desta técnica ser aplicada após a detecção dos olhos pelo gradiente (como vimos na seção 3.3.1). A distância entre os olhos foi utilizada para determinar a diferença de escala entre a imagem das características na face e nos exemplos obtidos como treinamento, considerando orientação da face aproximadamente frontal.

Outros fatores como iluminação, variações da característica facial e ruídos poderiam interferir no resultado da detecção por correlação. A primeira solução que adotamos foi aplicar uma equalização de histograma. Esta solução não apresentou melhorias nos resultados da detecção. Adotamos, então, uma segunda solução que foi utilizar, ao invés da imagem diretamente, o gradiente da imagem. Desta forma, a característica facial seria descrita não mais simplesmente por uma imagem, mas pela variação dos níveis de cinza entre os pontos desta imagem. Esta solução melhorou significativamente os resultados da detecção.

A figura 3.15 mostra as janelas de busca $f(x, y)$ e o resultado da busca de características

por correlação.

Figura 3.15: (a) Regiões de busca de características por correlação, (b) Resultado da busca.

O problema desta solução é que ela é aplicada individualmente a cada característica. Se houver uma falha no rastreamento de uma característica, esta poderia ser corrigida se analisássemos também o resultado global da posição das características naquele quadro. Isto foi feito utilizando modelos deformáveis. Os modelos deformáveis permitem que nós classifiquemos o resultado da posição do conjunto de características como aceitável ou não e, caso negativo, identifiquemos qual característica está tornando o modelo inaceitável.

A próxima seção apresenta o conceito de modelos deformáveis e sua aplicação em nosso sistema.

3.4 Modelos Deformáveis

Nas seções anteriores, procuramos detectar características faciais que nos permitissem descrever melhor a face, mas esta busca foi feita de forma individual para cada característica. Alguns erros resultantes dessa busca poderiam ser facilmente identificados se considerarmos algumas relações entre as características. Na figura 3.16, o erro seria facilmente identificado se comparássemos o resultado com um modelo da posição das características na face, já que

um dos cantos da boca se encontra visivelmente longe de sua posição esperada em um modelo de face.

Cootes e Taylor [CT04] propuseram um método para representar a face considerando a sua variabilidade de formas. Este método, conhecido como Modelos de Forma Ativa ou ASM (*Active Shape Model*), foi posteriormente estendido para inclusão de textura, o que originou os Modelos de Aparência Ativa ou AAM (*Active Appearance Model*). Estes modelos e suas variações são chamados de modelos estatísticos de forma ou modelos deformáveis.

Figura 3.16: Exemplo de erro da busca por correlação.

Nesta seção, apresentamos os princípios de modelos deformáveis. Ao final, descreveremos de que forma ele foi empregado em nosso sistema.

Modelo de forma

A forma de um objeto é modelada por um conjunto de pontos de controle, que podem ter qualquer dimensão. Em nosso caso, como estamos trabalhando com imagens bidimensionais, modelamos a face com pontos 2D.

A escolha dos pontos de controle é uma tarefa importante, pois contribui tanto para o desempenho do algoritmo (pontos que são facilmente identificados em diferentes imagens) quanto

para a descrição do objeto (pontos relevantes do objeto, que descrevem suas características mais importantes para outros algoritmos).

Cada forma é descrita por um vetor de $n \times d$ elementos, onde n é o número de pontos e d a dimensão dos pontos. Para isso, concatenamos os elementos dos vetores de cada posição individual dos pontos. Por exemplo, uma forma $\mathbf{x}$ formada por um conjunto de pontos (x_i, y_i) é representada por:

$$\mathbf{x} = (x_1, \dots, x_n, y_1, \dots, y_n)^T$$

Para criar um modelo de um objeto deformável, precisamos de um conjunto de treinamento que capte essa deformação através de diversos exemplos do objeto modelado.

Seja o conjunto de treinamento composto por s imagens. Geramos, manualmente, os vetores $\mathbf{x}_j, j = 1 \dots s$ que representam a forma do objeto em cada imagem. O próximo passo consiste em remover a variação entre estas formas que poderia ser atribuída às transformações de similaridade, como rotação, translação e escala. Assim teremos todas as formas representadas no mesmo sistema de coordenadas.

Alinhamento do conjunto de treinamento

Para alinhar o conjunto de treinamento, precisamos de um algoritmo que, aplicando transformações de similaridade em cada forma $\mathbf{x}_i$, minimize a distância desta forma à forma média $\bar{\mathbf{x}}$. Ou seja, queremos minimizar:

$$D = \sum |\mathbf{x}_i - \bar{\mathbf{x}}|^2 \quad (3.21)$$

Uma solução iterativa para este problema é a seguinte:

1. Tomamos cada forma e a transladamos para o seu centro de gravidade.

2. Tomamos uma forma como a estimação inicial da forma média e a escalonamos de forma que $|\bar{\mathbf{x}}_0| = 1$.
3. A estimação de $\bar{\mathbf{x}}_0$ define o sistema de coordenadas padrão.
4. Alinhamos as outras formas de acordo com a estimação atual da forma média.
5. Reestimamos a forma média a partir das formas alinhadas.
6. Aplicamos alguns limitantes na forma média estimada no passo anterior, alinhando-a com $\bar{\mathbf{x}}_0$ e escalonando-a de forma que $|\bar{\mathbf{x}}| = 1$.
7. Se não houve convergência, retornamos ao passo 4.

O alinhamento do passo 4 consiste em aplicar uma transformação de similaridade em cada forma, minimizando D . Como todas as formas tem seu centro de gravidade na origem, calculamos somente a variação de escala e orientação da forma.

A convergência ocorre quando a estimação da média não apresenta alteração significativa após uma iteração do algoritmo.

Modelando variação de forma

A variação de forma é modelada utilizando Análise de Componentes Principais (PCA). Aplicando PCA às formas $\mathbf{x}_i$, obtemos uma matriz Φ contendo os autovetores com maiores autovalores associados. Podemos, então, aproximar uma forma do conjunto de treinamento por:

$$\mathbf{x} = \bar{\mathbf{x}} + \Phi \mathbf{b}, \quad (3.22)$$

onde $\mathbf{b}$ é o vetor de parâmetros do modelo deformável, dado por:

$$\mathbf{b} = \Phi^T(\mathbf{x} - \bar{\mathbf{x}}). \quad (3.23)$$

Podemos gerar exemplos similares aos do conjunto de treinamento alterando o vetor de parâmetros $\mathbf{b}$. Mas, para que os exemplos gerados sejam realmente similares ao objeto modelado, precisamos aplicar limitantes em $\mathbf{b}$. Uma forma de se fazer isto é aplicar um limitante em cada elemento b_i , como por exemplo:

$$|b_i| \leq 3\sqrt{\lambda_i}. \quad (3.24)$$

A forma $\mathbf{x}$ relacionada a este vetor de parâmetros é considerada plausível.

Dada uma forma $\mathbf{x}'$, queremos encontrar a forma plausível $\mathbf{x}$ mais próxima de $\mathbf{x}'$.

Primeiramente, projetamos a forma no espaço dos modelos deformáveis, obtendo:

$$\mathbf{b} = \Phi'(\mathbf{x}' - \bar{\mathbf{x}}). \quad (3.25)$$

Se a forma $\mathbf{x}'$ não for plausível, aplicamos o limitante dado pela equação 3.24 até chegarmos a uma forma plausível. A partir destes parâmetros, obtemos a nova forma $\mathbf{x}$ através da equação 3.22.

Ajustando o modelo a novos pontos

Um modelo numa imagem pode ser descrito não somente pelos parâmetros de forma, mas também por transformações de similaridade como rotação, translação e escala. A posição dos pontos do modelo pode ser descrita da seguinte maneira:

$$\mathbf{x} = T_{X_t, Y_t, s, \theta}(\bar{\mathbf{x}} + \Phi\mathbf{b}), \quad (3.26)$$

onde T é uma transformação de similaridade envolvendo rotação (θ), escala (s) e translação (X_t, Y_t).

Dado um conjunto de pontos $\mathbf{Y}$, queremos encontrar a transformação T e os parâmetros $\mathbf{b}$ que minimizam a expressão:

$$|\mathbf{Y} - T_{X_t, Y_t, s, \theta}(\bar{\mathbf{x}} + \Phi \mathbf{b})|^2, \quad (3.27)$$

ou seja, queremos encontrar a forma plausível mais próxima da forma $\mathbf{Y}$, considerando que a forma pode sofrer transformações de similaridade.

Uma solução iterativa consiste dos seguintes passos:

1. Iniciamos os parâmetros do modelo, o vetor $\mathbf{b}$, com zero;
2. Geramos o modelo de instância dado pela forma média mais o resíduo: $\mathbf{x} = \bar{\mathbf{x}} + \Phi \mathbf{b}$;
3. Encontramos os parâmetros da transformação de similaridade que melhor mapeiam $\mathbf{x}$ para $\mathbf{Y}$;
4. Invertemos estes parâmetros e aplicamos a $\mathbf{Y}$, obtendo uma instância da forma $\mathbf{Y}$ no sistema de coordenadas do modelo: $\mathbf{y} = T^{-1}(\mathbf{Y})$;
5. Escalonamos $\mathbf{y}$ por $1/(\mathbf{y} \cdot \bar{\mathbf{x}})$;
6. Obtemos uma nova estimativa dos parâmetros de forma (utilizando a equação 3.23);
7. Aplicamos limitações em $\mathbf{b}$ conforme equação 3.24;
8. Se não convergiu, voltamos ao passo 2.

Novamente, a convergência é obtida quando não temos mudança significativa nos valores de $\mathbf{b}$.

No passo 3, precisamos encontrar os parâmetros de similaridade que melhor mapeiam $\mathbf{x}$ para $\mathbf{Y}$. Considerando que ambas as formas possuem centro de gravidade na origem, precisamos minimizar:

$$|sA\mathbf{x} - \mathbf{Y}|, \quad (3.28)$$

onde A é uma matriz de rotação por θ , e s a escala. Seja $\mathbf{x} = (x_1, \dots, x_n, y_1, \dots, y_n)^T$, $\mathbf{Y} = (x'_1, \dots, x'_n, y'_1, \dots, y'_n)^T$ e:

$$p = (\mathbf{x} \cdot \mathbf{Y})/|\mathbf{x}|^2,$$

$$q = \left(\sum_{i=1}^n n(x_i y'_i - y_i x'_i) \right) / |\mathbf{x}|^2.$$

Então teremos a escala dada por $s^2 = p^2 + q^2$ e o ângulo de rotação dado por $\theta = \tan^{-1}(p/q)$. A translação é dada pelo vetor formado pelos pontos do centro de gravidade de $\mathbf{x}$ e $\mathbf{Y}$. A demonstração da solução deste problema de minimização encontra-se em [CT04].

Busca de formas

Nas seções anteriores, descrevemos algoritmos capazes de detectar os olhos. Utilizamos a posição dos olhos para posicionar o modelo de faces. A partir do modelo médio de face, obtemos uma estimativa da posição inicial das demais características e fazemos uma busca por correlação ao redor desta posição inicial.

Aplicamos, então, o algoritmo descrito na seção 3.4 para atualizar a posição e os parâmetros de forma que melhor ajustam os pontos ao modelo de face. Ao redor dos pontos resultantes fazemos uma nova busca por correlação, repetindo o processo até obter convergência entre

os pontos obtidos por correlação e os pontos corrigidos pelo modelo de forma. Este processo iterativo é chamado de busca por Modelos de Forma Ativa ou ASM (*Active Shape Model*).

A figura 3.17 mostra algumas iterações deste algoritmo. A primeira figura mostra o resultado da detecção de características contendo um erro na detecção de um dos cantos da boca. A medida que aplicamos iterativamente os Modelos de Forma Ativa e a busca por correlação, este erro é detectado e corrigido até que torne o conjunto de pontos um modelo aceitável de face.

Figura 3.17: Iterações do algoritmo de busca por ASM.

3.5 Rastreamento de faces e características faciais

Antes de falarmos propriamente sobre rastreamento, precisamos definir o conceito de *tempo real*. Uma tarefa qualificada como tempo real deve possuir algumas características fundamentais [Encl]. Primeiramente, deve ter fluxo contínuo, ou seja, deve periodicamente receber dados em sua entrada, processá-los e gerar o resultado deste processamento. Segundo, as etapas de processamento dos dados e exibição dos resultados não devem afetar o fluxo de entrada de dados.

Algumas tarefas podem requerer um tempo máximo de atraso entre a entrada e a saída dos dados, outras podem simplesmente requerer que a saída dos dados aconteça antes da entrada de novos dados.

Em nosso caso, a entrada de dados consiste de uma sequência de vídeo, o processamento consiste da detecção da face e características faciais em cada quadro, e a saída consiste na renderização do quadro mostrando o resultado da detecção.

A maioria das câmeras de vídeo digital trabalham com uma taxa de 30 quadros por segundo. Desta forma, adotamos como meta para qualificar o nosso sistema de rastreamento em tempo real, a capacidade de processar um quadro e renderizar o resultado em 0.033 segundos.

Poderíamos, inicialmente, simplesmente aplicar os métodos descritos anteriormente e verificar se eles atingem a meta de 30 quadros por segundo. Mas mesmo que insistíssemos na simples otimização dos métodos, não estaríamos nos certificando de que faces identificadas em quadros adjacentes correspondem à mesma face.

As técnicas de rastreamento utilizam mais uma informação que é a posição da face no quadro anterior. Esta informação contribui para aumentar a certeza de que a face identificada em um determinado quadro é a mesma identificada no quadro anterior, além de otimizar o algoritmo, limitando o seu espaço de busca.

A seguir, apresentamos a técnica de rastreamento utilizada em nosso sistema, conhecida

como CAMShift [CM97]. Esta é uma técnica iterativa que se utiliza da informação de uma componente da cor do objeto para rastreamento.

CAMShift

O CAMShift (Continuously Adaptive Mean SHIFT) é um método descrito por Comaniciu e Meer [CM97], que utiliza informação de cor e a posição prévia do objeto para rastreamento.

O algoritmo tem como entrada um histograma que descreve o objeto a ser rastreado e a posição inicial deste na imagem. Esta posição inicial pode ser indicada manualmente, ou através de outras técnicas de detecção.

O histograma corresponde ao modelo de distribuição de densidade do objeto. Para reduzir o custo computacional do algoritmo, utiliza-se o histograma de somente uma componente cromática deste objeto, geralmente a matiz, que pode ser a componente H do espaço cromático HSV, ou a componente T do espaço TSL.

Dados uma imagem (por exemplo, um quadro de uma seqüência de vídeo) e o histograma de um determinado objeto a ser rastreado, criamos uma nova imagem a partir da inicial utilizando o histograma como tabela de consulta. Esta nova imagem é chamada *back-projection*, e indica a distribuição de probabilidade do modelo do objeto na imagem.

O algoritmo então faz uma busca iterativa pelo objeto na área ao redor da posição anterior do objeto. O resultado da busca é utilizado como entrada para o próximo processo de detecção. A figura 3.18 mostra os passos deste algoritmo.

Embora eficiente e robusto, este algoritmo é capaz de rastrear somente um objeto por vez. Por fazer uma análise somente por cor, é facilmente enganado por outros objetos de mesma cor que se aproximem da região rastreada, como por exemplo quando duas faces se aproximam.

A figura 3.19 mostra o resultado da aplicação do CAMShift para uma seqüência de imagem obtida de uma câmera Sony EVI-D30 e um computador Pentium 4 2.8GHz, obtendo um

Figura 3.18: CAMShift

processamento de 30 quadros por segundo.

Embora esta técnica tenha se mostrado robusta e eficiente para rastrear faces, ela não é aplicável ao rastreamento de características faciais, já que a cor não é uma propriedade suficiente para a descrição de algumas características da face essenciais no processo de estimar a sua orientação. Por isso, decidimos utilizar outra técnica baseada em correlação para rastrear as características faciais que apresentaremos a seguir.

Figura 3.19: Rastreamento com CAMShift

Rastreamento de características faciais

O método para rastrear características faciais parte do mesmo princípio do rastreamento da face: o conhecimento da posição prévia da característica rastreada.

A busca no primeiro quadro se dará pelos métodos descritos na seção 3.3. Nos quadros seguintes, a busca pela nova posição da característica é feita por correlação aplicada somente ao redor de uma janela em torno da posição da característica no quadro anterior. Filtramos o resultado privilegiando os pontos mais próximos da posição detectada no quadro anterior.

Capítulo 4

Resultados Experimentais e Aplicações

Neste capítulo apresentamos os resultados experimentais obtidos pelo sistema e descrevemos duas aplicações desenvolvidas a partir deste.

Baseado na proposta inicial de nossa pesquisa, desenvolvemos um sistema capaz de detectar a face bem como algumas características faciais como os olhos (centro da íris), nariz (narinas) e boca (cantos da boca). A entrada do sistema é composta por uma imagem ou seqüência de vídeo digital, que pode ser um arquivo ou entrada de uma câmera. No caso de vídeo, o sistema executa o rastreamento da face e suas características.

O resultado da detecção das características faciais serve de entrada para o algoritmo de estimação da orientação da face.

4.1 Resultados da detecção de cor de pele

Para o aprendizado da cor de pele, utilizamos como treinamento amostras de um banco de dados de face constituído de pessoas de diferentes etnias. Utilizamos 15 imagens de face, sendo 3 morenos, 2 negro e 10 brancos. Extraímos manualmente regiões de pele e criamos uma base

de dados de cor de pele que serviu de treinamento conforme descrito na seção 3.1.

Figura 4.1: Imagens de face utilizadas para treinamento de cor de pele.

A figura 4.1 mostra imagens do banco de dados utilizado para este treinamento. Estas imagens foram geradas utilizando somente regiões de cor de pele das amostras de faces, não sendo utilizadas regiões de boca, olhos e cabelo.

A figura 4.2 mostra alguns resultados da detecção de cor de pele. Como podemos ver, o sistema apresenta bons resultados na detecção de regiões de cor de pele. Na terceira figura, porém, vemos que o cabelo da mulher teve um resultado mais intenso de detecção do que a própria pele. Além disso, outros objetos de cromaticidade próxima a cromaticidade da cor de pele geraram resultados falso-positivos.

Figura 4.2: Resultados da detecção de cor de pele

4.2 Resultados da detecção de características

Para detectar características faciais, implementamos dois algoritmos com o objetivo inicial de detectar os olhos: a transformada de Hough para detecção de círculos e o algoritmo de detecção pelo vetor gradiente.

Os pontos selecionados por esses algoritmos como melhores candidatos são pareados aplicando heurísticas simples conforme descrito na seção 3.3.1. As demais características são detectadas por correlação com base em um modelo médio da face.

A figura 4.3 mostra alguns resultados da detecção de características faciais utilizando o algoritmo de detecção pelo vetor gradiente.

Também utilizamos os Modelos de Forma Ativa para verificar e corrigir falhas na detecção das características, ajustando o conjunto de características detectadas de acordo com um modelo deformável da face.

Figura 4.3: Resultados da detecção de características faciais utilizando o algoritmo de detecção pelo gradiente para detecção de olhos e correlação para as demais características.

Para verificar se os resultados obtidos pela detecção de face e características faciais correspondem a uma face, aplicamos o método de PCA somente numa região interna da face, que inclui olhos, nariz e boca.

O grande problema da aplicação do método PCA para verificação de faces ocorre na etapa de comparação da imagem de entrada com a imagem projetada no espaço de faces. Quando a face de entrada faz parte do conjunto de treinamento, a projeção desta resulta numa face similar à entrada. Quando, porém, se trata de uma nova face, a projeção desta nova face resulta em outra face geralmente diferente da face de entrada. Isto torna difícil determinar um limiar adequado que separasse satisfatoriamente os resultados positivos e negativos. Optamos, então, por escolher a região que apresenta menor diferença entre a imagem de entrada e sua projeção no espaço de faces.

O algoritmo de PCA também foi utilizado para reconhecimento de face em uma ferramenta de indexação de vídeo implementada em nosso laboratório. Deste trabalho resultou um artigo publicado no Workshop de Teses e Dissertações do SIBGRAPI 2005 [BSSM05]. Esta ferramenta será apresentada na seção 4.3.1.

4.3 Aplicações

A fim de demonstrar a eficácia dos métodos desenvolvidos, apresentamos nesta seção duas aplicações da detecção de faces e características faciais. A primeira aplicação é uma ferramenta de indexação de vídeo que permite organizar e realizar consultas a um repositório de dados de vídeo baseado no conteúdo das imagens. A segunda aplicação é um estimador da orientação de face utilizando um modelo simples de face e aplicando um algoritmo baseado em geometria projetiva.

4.3.1 Indexação em vídeo

O crescimento do mercado por equipamento de imagem digital possibilitou a criação de um grande volume de dados de imagens fotográficas e vídeo em um curto período de tempo. Por outro lado, este grande volume de dados dificulta a sua organização, o que impossibilita a manipulação e consulta eficiente a estes dados. Isto tornou fundamental o desenvolvimento de ferramentas interativas que possibilitem consultas baseadas na informação visual destes dados.

Atualmente, consultas a um repositório de fotos e vídeos digitais é baseada em informação anotada manualmente. Além de exigir um grande esforço humano, esta informação é geralmente incompleta e imprecisa, considerando a vasta possibilidade de consultas que podemos fazer a este repositório. Uma forma de expressar informação visual é através do uso de padrões na forma de imagens, ou *sketches*. Outra estratégia é o uso de filtros para detecção de objetos específicos nas imagens. Utilizando esta última alternativa, desenvolvemos uma ferramenta de indexação de vídeo digital baseado na ocorrência de faces humanas.

A ferramenta de indexação de vídeo é composta de três módulos. O primeiro módulo consiste no reconhecimento de faces. Primeiramente a face é detectada utilizando cor de pele e o detector de olhos descritos no capítulo 3. O PCA, descrito na seção 3.2, é utilizado então

para verificação e reconhecimento da face.

Para reconhecer faces utilizando PCA, cada indivíduo é identificado por uma classe que é representada pelo vetor de pesos $\Omega^T = [\omega_1, \omega_2, \dots, \omega_M]$, definido na equação 3.9, correspondente à projeção da imagem de face do indivíduo no espaço de faces. Dada uma nova face, esta é projetada no espaço de faces e o seu vetor de pesos é comparado com as classes disponíveis. Se o vetor de pesos é similar a alguma das classes disponíveis, então esta face é identificada como sendo do indivíduo correspondente à classe escolhida. Para uma seqüência de vídeo, construímos uma tabela de faces onde, para cada quadro do vídeo, informamos se existe algum indivíduo conhecido presente naquele quadro.

O segundo módulo consiste na segmentação de arquivos de vídeo em tomadas. Uma tomada pode ser definida como uma seqüência contínua de quadros relacionados, separada por transições. As transições podem ocorrer em forma de cortes, que são simples concatenações de tomadas, ou transições graduais, como os efeitos de *fades* e *dissolves*, em que o conteúdo visual altera-se progressivamente entre tomadas consecutivas. A segmentação dos arquivos de vídeo é feita através de um algoritmo de detecção de fronteiras de tomadas (*Shot-boundary detection*), que consiste na detecção de todas as possíveis transições existentes numa seqüência de vídeo. Mais detalhes deste algoritmo podem ser vistos em [San04].

O terceiro módulo consiste na interface gráfica da ferramenta, como vemos na figura 4.4. A interface é composta de três regiões. Na região inferior temos o navegador pelo índice de conteúdo do repositório de imagens e vídeos digitais, onde os vídeos são segmentados em tomadas. Na região superior direita temos um *player*, que permite a execução instantânea de uma tomada selecionada a partir do índice de conteúdo. Na região superior esquerda, temos um índice de faces conhecidas pelo nosso sistema. Em nosso exemplo, temos apenas uma face. Ao clicar na face de um indivíduo deste índice, o sistema destaca com uma borda vermelha todas as tomadas em que este indivíduo aparece.

Figura 4.4: Interface da ferramenta de indexação em vídeo.

Como teste, utilizamos uma sequência de vídeo de aproximadamente 30 minutos de um noticiário da *ABC World News Tonight*. O vídeo é composto de 51015 quadros, que foram separados em 427 tomadas. Em 18 delas, estava presente o âncora do noticiário, que é um indivíduo conhecido pelo classificador de PCA, e portanto presente no índice de faces. Para treinar o classificador de PCA, utilizamos 20 imagens do âncora extraídas manualmente de diferentes quadros do vídeo. Quando um âncora estava presente numa tomada, em geral ele estava presente em todos os quadros daquela tomada, em diferentes orientações e escala.

Limiar	VP	FN	VN	FP	Precisão	Revocação
10%	94%	6%	76%	24%	14.78%	94%
20%	89%	11%	83%	17%	18.60%	89%
30%	89%	11%	87%	13%	22.53%	89%
40%	72%	28%	90%	10%	23.63%	72%
50%	67%	33%	93%	7%	29.27%	67%

Tabela 4.1: Resultados do desempenho do sistema de indexação em vídeo, VP: verdadeiro-positivo, FN: falso-negativo, VN: verdadeiro-negativo, FP: falso-positivo.

Para dizer que um indivíduo está presente em uma tomada, ele deve estar presente em pelo menos uma certa porcentagem dos quadros desta tomada. A escolha de um limiar para esta porcentagem interfere diretamente no desempenho do sistema. Quanto menor o limiar escolhido, menor o número de falso-negativos, porém maior o número de falso-positivos.

A tabela 4.1 mostra o resultado do desempenho do sistema. Nas duas últimas colunas desta tabela temos as medidas de precisão e revocação dos resultados [BC94]. Estas são medidas comumente utilizadas na área de recuperação de informação, responsável por avaliar a proporção de falso-positivos e falso-negativos na solução proposta. A medida de precisão (P) é dada pela razão entre o número de verdadeiro-positivos (VP) e a soma entre os resultados verdadeiro-positivos e verdadeiro-negativos (VN), e a medida de revocação (R) é dada pela razão entre o número de verdadeiro-positivos (VP) e a soma entre os resultados verdadeiro-positivos e falso-negativos (FN).

$$P = \frac{VP}{VP + VN} \times 100\% \quad (4.1)$$

$$R = \frac{VP}{VP + FN} \times 100\% \quad (4.2)$$

Quando aumentamos o limiar para a porcentagem de quadros que contém o âncora, estamos

exigindo que o algoritmo detecte este indivíduo em um maior número de quadros da tomada. Com isto, temos uma diminuição dos resultados verdadeiro-negativos, mas por outro lado, temos um prejuízo nos resultados verdadeiro-positivos. Logo, como vemos pela tabela, temos que a precisão aumenta e a revocação diminui conforme aumentamos este limiar.

Uma das principais dificuldades notadas neste aplicativo ocorre quando o algoritmo de detecção de cor de pele falha. Neste caso, como não temos as regiões de face corretamente localizadas, PCA não é capaz de detectar nenhuma face, aumentando o número de falso-negativos. Outra dificuldade consiste em alinhar a face ao modelo de PCA. Os ruídos e o tamanho reduzido da face na maioria dos quadros do vídeo tornam muito impreciso os resultados de detecção das características faciais, o que nos obriga a diminuir o limiar de PCA para detectar e reconhecer faces, aumentando o número de falso-positivos.

Uma solução seria aplicar algoritmos de rastreamento nas faces detectadas. Desta maneira, se conseguíssemos manter o rastreamento de uma mesma face identificada por um certo número de quadros da tomada, definiríamos que o indivíduo está presente nesta tomada. Esta solução poderá ser aplicada em experimentos futuros.

4.3.2 Estimação da Orientação da Face

Outra aplicação da detecção da face e características faciais é o seu uso para estimar a orientação da face.

Embora a face não seja um objeto rígido, o método proposto por DeMenthon e Davis [DD95] para determinar a orientação de um objeto rígido numa imagem tem sido também utilizado em algoritmos de estimação de orientação da face [CT02].

A orientação da face pode ser estimada a partir de um modelo rígido tridimensional das características faciais. Utilizando algoritmos de geometria projetiva e a posição bidimensional destas características na imagem, obtemos a orientação que melhor projeta o modelo tridi-

mensional da face nesta imagem.

Este método, baseado em técnicas de álgebra linear, é composto por dois algoritmos, POS e POSIT. Ele assume um modelo do objeto e a detecção de pelo menos 4 pontos não-coplanares do objeto na imagem. O modelo do objeto consiste de um conjunto de pontos que determinam a localização de pontos de características do objeto identificáveis na imagem.

O primeiro algoritmo, conhecido como POS (Pose from Orthography and Scaling), faz uma estimativa aproximada da posição do objeto na imagem utilizando produtos de uma matriz por dois vetores. A matriz contém informações de pontos do modelo do objeto, e os vetores contêm informações das coordenadas destes pontos na imagem. Como resultado, temos dois vetores que, normalizados, constituem as primeiras linhas da matriz de rotação, e suas normas correspondem ao fator de escala da projeção, que provê o vetor de translação.

O segundo algoritmo consiste das mesmas operações, aplicadas com os pontos da imagem obtidos no passo anterior. A cada iteração obtemos uma estimativa mais precisa da orientação do objeto. Por isso, este segundo algoritmo é chamado POSIT (POS com ITerações).

Notação

A câmera tem centro de projeção O e os seus eixos de coordenadas são representados pelos vetores unitários $\mathbf{i}$ e $\mathbf{j}$ correspondendo às linhas e colunas da imagem, e $\mathbf{k}$ correspondendo ao eixo óptico. A imagem está a uma distância f do centro de projeção O .

O objeto tem pontos de características $M_0, \dots, M_i, \dots, M_n$. O sistema de coordenadas do objeto é centrado em M_0 e seus eixos são u, v, w . As coordenadas (U_i, V_i, W_i) do ponto M_i são conhecidas no sistema de coordenadas do objeto, e desconhecidas no sistema de coordenadas da câmera.

A projeção perspectiva dos pontos M_i no plano da imagem nos fornece os pontos m_i com coordenadas (x_i, y_i) .

Figura 4.5: Sistema de Coordenadas da Câmera e Objeto [DD95]

Definição do problema

O objetivo deste método é encontrar a matriz de rotação, que permitirá transformar vetores definidos no sistema de coordenadas do objeto em vetores definidos no sistema de coordenadas da câmera, e o vetor de translação do objeto.

Verificamos que o produto interno $\langle \mathbf{M}_0 \mathbf{M}_i, \mathbf{i} \rangle$, ambos definidos no sistema de coordenadas do objeto, resulta na projeção do vetor $\mathbf{M}_0 \mathbf{M}_i$ sobre o vetor unitário $\mathbf{i}$. O mesmo se aplica à $\mathbf{j}$ e $\mathbf{k}$.

As linhas da matriz de rotação $\mathbf{R}$ consistem, então, dos vetores $\mathbf{i}$, $\mathbf{j}$ e $\mathbf{k}$ expressos no sistema de coordenadas do objeto.

$$\mathbf{R} = \begin{bmatrix} i_u & i_v & i_w \\ j_u & j_v & j_w \\ k_u & k_v & k_w \end{bmatrix} \quad (4.3)$$

Observe que o vetor $\mathbf{k}$ pode ser obtido do produto vetorial entre $\mathbf{i}$ e $\mathbf{j}$.

O vetor de translação $\mathbf{T}$ é o vetor entre as origens dos sistemas de coordenadas da câmera e do objeto, ou seja, $\mathbf{OM}_0$. Se $\mathbf{M}_0$ é um ponto visível na imagem, então temos as coordenadas de $\mathbf{m}_0$ e o vetor $\mathbf{T}$ pode ser escrito como:

$$\mathbf{T} = \mathbf{OM}_0 = \frac{Z_0}{f} \mathbf{Om}_0 \quad (4.4)$$

Projeção Perspectiva e Projeção Escalar Ortográfica

A projeção perspectiva dos pontos M_i no plano G da imagem nos fornece os pontos m_i com coordenadas:

$$x_i = \frac{f}{Z_i} X_i \quad \text{e} \quad y_i = \frac{f}{Z_i} Y_i. \quad (4.5)$$

A projeção ortográfica escalar (*SOP, Scaled Orthographic Projection*) consiste de uma aproximação para a projeção perspectiva, assumindo a mesma profundidade Z_i para os diferentes pontos M_i , como por exemplo, a profundidade Z_0 . Ao invés de m_i teríamos p_i com coordenadas:

$$x'_i = \frac{f}{Z_i} X_i \quad \text{e} \quad y'_i = \frac{f}{Z_i} Y_i. \quad (4.6)$$

O ponto M_0 tem as mesmas coordenadas em SOP e na projeção perspectiva. Na figura 4.5 observamos a construção geométrica de SOP. Seja o plano K passando por M_0 paralelo ao

plano G . O ponto M_i é projetado em K por uma projeção ortográfica, resultando no ponto P_i , que, por sua vez, é projetado no plano G por uma projeção perspectiva, resultando em p_i . Temos, então, o vetor m_0p_i paralelo a M_0P_i .

Podemos reescrever as coordenadas de SOP como:

$$x'_i = sX_0 + s(X_i - X_0) = x_0 + s(X_i - X_0) \quad (4.7)$$

$$y'_i = y_0 + s(Y_i - Y_0) \quad (4.8)$$

onde $s = \frac{f}{Z_0}$ é o fator de escala.

Esta equação mostra a proporcionalidade entre m_0p_i e M_0P_i .

Equações Fundamentais

Para definir a orientação do objeto, precisamos calcular as coordenadas de $\mathbf{i}$ e $\mathbf{j}$ no sistema de coordenadas do objeto para determinar a matriz de rotação, e a coordenada Z_0 para determinar o vetor de translação. Faremos isto utilizando as coordenadas de M_0M_i e as coordenadas (x_i, y_i) dos pontos m_0 e m_i da seguinte maneira:

$$M_0M_i \frac{f}{Z_0} \mathbf{i} = x_i(1 + \varepsilon_i) - x_0, \quad (4.9)$$

$$M_0M_i \frac{f}{Z_0} \mathbf{j} = y_i(1 + \varepsilon_i) - y_0 \quad (4.10)$$

$$\text{onde } \varepsilon_i = \frac{1}{Z_0} M_0M_i.k$$

A demonstração destas equações está em [DD95]. Os termos $x_i(1 + \varepsilon_i)$ e $y_i(1 + \varepsilon_i)$ são, na

verdade, as coordenadas x'_i e y'_i dos pontos p_i , como podemos ver abaixo:

$$\mathbf{M}_0\mathbf{M}_i = \mathbf{M}_0\mathbf{P}_i + \mathbf{P}_i\mathbf{M}_i \quad (4.11)$$

$$= \frac{Z_0}{f}\mathbf{m}_0\mathbf{p}_i + \mathbf{P}_i\mathbf{M}_i \quad (4.12)$$

Tomando o produto interno $\langle \mathbf{M}_0\mathbf{M}_i, \mathbf{i} \rangle$, do lado direito teremos a soma dos produtos internos $\langle \mathbf{m}_0\mathbf{p}_i, \mathbf{i} \rangle$ que resulta na coordenada x do vetor $\mathbf{m}_0\mathbf{p}_i$, que é dada pela diferença $x'_i - x_0$, e $\langle \mathbf{P}_i\mathbf{M}_i, \mathbf{i} \rangle$ que resulta em zero, pois os vetores são ortogonais entre si. Temos, então:

$$\mathbf{M}_0\mathbf{M}_i \cdot \mathbf{i} = \frac{Z_0}{f}\mathbf{m}_0\mathbf{p}_i \cdot \mathbf{i} + \mathbf{P}_i\mathbf{M}_i \cdot \mathbf{i} \quad (4.13)$$

$$= \frac{Z_0}{f}(x'_i - x_0) + 0; \quad (4.14)$$

$$\mathbf{M}_0\mathbf{M}_i \cdot \frac{f}{Z_0}\mathbf{i} = x'_i - x_0 \quad (4.15)$$

Algoritmo POS e POSIT

As equações fundamentais (4.9) podem ser reescritas como:

$$\mathbf{M}_0\mathbf{M}_i \cdot \mathbf{I} = x_i(1 + \varepsilon_i) - x_0, \quad (4.16)$$

$$\mathbf{M}_0\mathbf{M}_i \cdot \mathbf{J} = y_i(1 + \varepsilon_i) - y_0 \quad (4.17)$$

$$\text{onde } \mathbf{I} = \frac{f}{Z_0}\mathbf{i} \text{ e } \mathbf{J} = \frac{f}{Z_0}\mathbf{j}$$

Se tivermos um valor para ε_i , então os únicos valores desconhecidos das equações acima são os valores de $\mathbf{I}$ e $\mathbf{J}$. Ao calcularmos e normalizarmos estes vetores obtemos $\mathbf{i}$ e $\mathbf{j}$, e a norma de $\mathbf{I}$ e $\mathbf{J}$ correspondem à terceira coordenada do vetor de translação, T_z .

Como o valor de ε_i é desconhecido, inicialmente atribuímos zero. Neste caso, estamos assumindo uma projeção ortográfica do objeto, e portanto obteremos somente uma aproximação de $\mathbf{i}$ e $\mathbf{j}$. Estes cálculos correspondem ao algoritmo POS (*Pose from Orthography and Scaling*).

Esta aproximação nos permite computar valores mais exatos de ε_i , pois obtemos $\mathbf{k}$ a partir do produto vetorial de $\mathbf{i}$ e $\mathbf{j}$, e assim, iterativamente, aplicamos o algoritmo POS obtendo sempre valores mais precisos de $\mathbf{i}$, $\mathbf{j}$ e T_z . Este cálculo corresponde ao POSIT (POS com ITerações), já que consiste em iterações do POS, convergindo para um valor mais exato da orientação do objeto.

Nos experimentos feitos por DeMenthon e Davis, aproximadamente 5 iterações eram suficientes para se obter uma orientação precisa do objeto. Como podemos ver, entre as vantagens deste algoritmo estão a sua eficiência, pois requer poucas operações aritméticas, tornando-o viável para uso em tempo-real; a facilidade de implementação, utilizando poucas linhas de código; e o fato de não requerer uma estimativa da orientação inicial.

A figura 4.6 mostra um exemplo de saída da aplicação do POSIT para estimar a orientação de face da imagem. As linhas azuis mostram o alinhamento do modelo de face com as características faciais obtidas pela aplicação do POSIT, e em vermelho temos o plano formado pelos olhos e cantos da boca e o vetor normal a este plano, indicando a orientação estimada pelo algoritmo.

Resultados

Para testar a estimativa da orientação de face, utilizamos as imagens criadas pelo grupo FGnet para o *Pointing'04 ICPR Workshop* [GILLIG]. Este conjunto de imagens é constituído por

Figura 4.6: Face com orientação de 58° no plano vertical e 58° no plano horizontal.

imagens de face de 15 pessoas, sendo 2 mulheres e 13 homens, em orientações verticais e horizontais de 0° , $\pm 15^\circ$, $\pm 30^\circ$, $\pm 45^\circ$, $\pm 60^\circ$ e $\pm 90^\circ$. Para cada indivíduo há dois conjuntos de imagens, um com o indivíduo no estado natural e outro com o indivíduo vestindo algum acessório como óculos ou variando a sua expressão facial.

Utilizamos, como modelo tridimensional da face, um conjunto formado por três pares de pontos: olhos (íris), narinas e cantos da boca. Para obter o modelo tridimensional das características faciais, primeiramente obtivemos o modelo bidimensional a partir da posição das características em imagens de faces frontais. Utilizando imagens de perfil (90°) do mesmo indivíduo, pudemos determinar a diferença de profundidade das características, obtendo assim um modelo tridimensional da face.

A figura 4.7 mostra um indivíduo sob vários ângulos, horizontais (H) e verticais (V). Como podemos observar, a partir de 45° já não temos a presença de todas as características faciais, como por exemplo, os olhos. Por isto, para nosso teste, utilizamos somente as imagens com orientação de 0° , $\pm 15^\circ$ e $\pm 30^\circ$ nas direções horizontal e vertical, totalizando 750 imagens, sendo 50 imagens de cada indivíduo. Nessas imagens, aplicamos o método de detecção de cor de pele e subtraímos as imagens em que o este método não foi capaz de identificar a região de

face corretamente, restando 599 imagens onde a face foi detectada com sucesso. Dessas, 255 imagens possuem orientação com variação máxima de 15° e 26 imagens são frontais.

Figura 4.7: Exemplos de entrada.

Para determinar se a característica foi corretamente detectada, a inspeção foi feita manualmente em cada imagem. No caso do olho, o ponto encontrado deveria estar dentro da região delimitada pela íris. No caso do nariz, os pontos deveriam estar na região escura das narinas quando estas aparecem, ou o mais próximo possível delas. No caso da boca, os pontos deveriam aparecer nos cantos extremos da boca.

A tabela 4.2 mostra a média de acertos da detecção dos olhos pelo método do gradiente, que corresponde ao número de imagens em que a característica foi corretamente detectada dividido pelo total de imagens utilizadas no teste. Observe que para faces frontais, o sistema detecta corretamente os olhos em 46% das imagens. Essa taxa é relativamente baixa devido a

presença de ruídos, como acessórios, variação de iluminação e variação de expressões faciais, que geram muitos candidatos falso-positivos na aplicação do método do gradiente. A medida que o ângulo da face aumenta, os olhos se tornam mais difíceis de detectar, pois as regras heurísticas utilizadas são mais adequadas para faces frontais, não sendo apropriadas quando temos variação de orientação que dificulte aproveitar a informação de simetria com relação à posição dos olhos, e quando, na aplicação do método de gradiente, a variação do tamanho da íris resulte em uma acumulação mais fraca de vetores gradiente apontando para um dos olhos.

A tabela 4.3 mostra a média de acertos da detecção das demais características dado que os olhos foram corretamente detectados. Observe que quando os olhos são corretamente detectados, as demais características são corretamente detectadas também (cerca de 92% de acerto). No entanto, quando a orientação da face é em torno de 15° , o desempenho do sistema já cai para 70%. Isso é devido principalmente a deslocação das características em relação à região esperada de ocorrência delas. Quando aumentamos a orientação da face até 30° , vemos que a tendência de queda do desempenho do sistema se mantém, indicando que os métodos aplicados são mais apropriados para imagens frontais.

Ângulo	Média
0°	0.46
15°	0.39
30°	0.26

Tabela 4.2: Média de acerto da detecção dos olhos. A primeira coluna indica o ângulo máximo de orientação da face das imagens.

A figura 4.8 mostra um gráfico das duas tabelas anteriores, que deixa claro que as técnicas apresentadas se tornam menos robustas a medida que aumentamos o grau de orientação da face. Isto era esperado devido ao fato de que as técnicas utilizadas neste trabalho foram criadas principalmente para tratar de imagens de faces frontais. Para ângulos maiores que

Ângulo	Média
0°	0.92
15°	0.70
30°	0.63

Tabela 4.3: Média de acerto da detecção das demais características faciais nas imagens em que os olhos foram corretamente detectados.

45° , algumas das características rastreadas já não podem ser vistas na imagem, e portanto o nosso método não pode ser aplicado.

Figura 4.8: Média de acertos da detecção de características faciais

A tabela 4.4 mostra o erro da estimação da orientação. Estes valores representam a média e o desvio-padrão da diferença entre o valor esperado e o valor estimado da orientação da face. Como vemos nesta tabela, a variação do erro não está ligada ao grau do ângulo de orientação da face.

Ângulo	Média	Desvio Padrão
0°	28.29	18.91
15°	22.22	15.47
30°	25.82	18.36

Tabela 4.4: Média e desvio padrão da diferença entre o valor esperado e o valor estimado da orientação da face.

Embora a diferença entre o valor esperado e o valor estimado da orientação da face seja alto, analisando visualmente as imagens, o POSIT respondeu corretamente com relação ao sentido da orientação da face na maioria dos casos, como podemos ver na figura 4.9, que possui diversos exemplos de saída do algoritmo.

Diversos parâmetros interferem nos resultados obtidos pelo sistema como, por exemplo, o limiar aplicado na detecção de cor de pele, os limiares dos filtros gradiente para detecção de olhos e os valores obtidos dos treinamentos executados pelo sistema para a aplicação dos algoritmos descritos.

Se optarmos por treinar o sistema e escolher os limiares adequados para um indivíduo específico, obtemos uma melhoria significativa nos resultados do sistema, como podemos ver nas tabelas 4.5, 4.6 e 4.7.

Ângulo	Média
0°	1
15°	0.89
30°	0.5

Tabela 4.5: Média de acerto da detecção dos olhos para um mesmo indivíduo.

A figura 4.10 mostra exemplos da saída do sistema treinado para um indivíduo.

Ângulo	Média
0°	1
15°	0.81
30°	0.86

Tabela 4.6: Média de acerto da detecção das demais características faciais nas imagens em que os olhos foram corretamente detectados.

Ângulo	Média	Desvio Padrão
0°	13.27	9.47
15°	11.48	5.28
30°	14.82	18.22

Tabela 4.7: Média e desvio padrão da diferença entre o valor esperado e o valor estimado da orientação da face de um mesmo indivíduo.

Figura 4.9: Exemplos de saída da estimação de orientação da face.

Figura 4.10: Exemplos de saída da estimação de orientação da face de uma mesma pessoa.

Capítulo 5

Comentários finais

Durante anos, o LaTIn tem desenvolvido pesquisas na área de Interação Humano Computador. Entre os seus principais trabalhos está o rastreador de pupilas, descrito na seção 2.1.2, que se utiliza de fontes de luz infra-vermelha ligadas alternadamente, produzindo pupilas brilhantes e escuras em quadros consecutivos, e a subtração destes quadros possibilita a detecção das pupilas. Por se utilizar destas fontes de luz, esta técnica não pode ser utilizada em imagens e sequência de vídeo digital.

Nossa principal motivação foi desenvolver uma nova técnica para detecção de faces e características faciais aplicável a imagens e vídeo digital. Partindo de imagens coloridas, primeiramente implementamos um método de detecção de cor de pele para identificar regiões candidatas à face. Embora pessoas de diferentes etnias possam apresentar diferentes cores de pele, vários estudos mostraram que essas variações são devidas apenas a quantidade de melanina, e que a cor de pele em si pode ser descrita utilizando apenas a informação de cromaticidade [YLAB98], sendo a luminância e a saturação responsáveis pela variação de cores de pele existentes.

Os métodos baseados em cor de pele são em geral bastante robustos e computacionalmente

eficientes. Dentre os vários espaços de cor sugeridos, a utilização da componente T do espaço cromático TSL mostrou-se robusta e capaz de identificar pessoas de diferentes etnias e em condições variadas de iluminação [IPBM04]. Além disso, a aplicação de métodos de erosão e dilatação nas regiões de cor de pele elimina ruídos e evita que uma mesma face seja identificada por regiões desconexas. Pelo contrário, as regiões candidatas que correspondem a uma face geralmente possuem a forma aproximada de uma face. Isto contribui para determinar a localização esperada das demais características faciais.

Uma vez detectada regiões com cor de pele candidatas a faces, nosso sistema busca por características faciais apenas nessas regiões. A primeira dessas características são os olhos, e para isso, desenvolvemos um método que denominamos de gradiente, por utilizar a diferença de intensidade luminosa entre a esclera e a íris. Embora computacionalmente caro, esse método é invariante a orientação da face, ao contrário de outros métodos tradicionais baseados em *template matching*.

A eficácia do método de detecção dos olhos pelo gradiente depende muito de alguns fatores como o tamanho dos olhos e o contraste entre a esclera e a íris. Algumas constantes precisam ser definidas, como o raio médio da região circular formada pela íris e os limiares para aplicação dos filtros de gradiente. Definimos estas constantes de acordo com o conjunto de imagens utilizadas no teste, de forma a otimizar os resultados. Futuramente, podemos estudar uma forma de obter estes valores dinamicamente a partir de informações previamente obtidas da face.

Para escolher os melhores candidatos a olhos, pareamos os pontos candidatos utilizando algumas regras heurísticas, e aplicamos PCA na região correspondente a cada par de olhos, com o objetivo de verificar qual candidato se aproximava mais de uma face. PCA não se mostrou muito robusto, mesmo com algumas medidas tomadas como o alinhamento da face às eigenfaces e a equalização do histograma tanto das imagens de treinamento como das imagens

de entrada do algoritmo. Uma medida a ser tomada é estudar métodos que minimizem os efeitos de iluminação. Outra possibilidade é a aplicação do PCA no domínio da frequência. Recentes estudos mostram grande melhoria de resultados da aplicação de PCA neste domínio utilizando somente o ângulo da fase para compor o espaço de faces, o que foi chamado de *eigenphases* [SKK01].

As demais características são localizadas por correlação. Para verificar se os resultados obtidos são consistentes com uma face real utilizamos os modelos deformáveis, que se mostraram uma técnica poderosa para ajuste de modelos complexos de face. Em nosso caso, porém, como utilizamos um modelo simples de face, composto por somente seis pontos, os modelos deformáveis pouco contribuíram para a correção de problemas na detecção de características, já que, desde que os olhos sejam detectados corretamente, somente o uso da correlação mostrou-se suficiente para a detecção das demais características. Como trabalho futuro, sugerimos a utilização de um modelo mais descritivo da face, composto por mais pontos descrevendo todo o contorno da face e suas características faciais.

No rastreamento de faces e características faciais, o filtro de Kalman seria um algoritmo mais apropriado para prever a posição das características em um determinado quadro. Ele se utiliza de um modelo de comportamento e da observação dos estados anteriores do sistema para prever o estado seguinte. Uma vantagem na utilização deste método é a sua capacidade de melhorar a previsão do estado do sistema à medida que ele atualiza o modelo de comportamento do objeto. Além disso, diferente do CAMShift em que a busca pelo objeto é feita somente pelo seu histograma, a processo de observação do objeto utilizando o filtro de Kalman é independente da previsão, permitindo a escolha do método mais adequado de verificação de acordo com o objeto rastreado.

Para testar o desempenho de nosso detector e rastreador de faces, desenvolvemos duas aplicações. A primeira é uma ferramenta de indexação de vídeo, criado com o objetivo de

facilitar a busca de personagens em bibliotecas multimídia. Atualmente, o sistema faz uma busca quadro a quadro pelos indivíduos de seu conjunto de treinamento. Como trabalho futuro sugerimos estender esta ferramenta utilizando melhor as informações de rastreamento, permitindo que um indivíduo seja identificado somente uma vez e depois rastreado até que ele saia da cena. Além disso, podemos extrair a orientação da face dos indivíduos, permitindo identificar até mesmo cenas de diálogo, com duas pessoas face a face.

A segunda aplicação que implementamos é um estimador da orientação da face utilizando o algoritmo POSIT, que se utiliza de um modelo tridimensional fixo da face e dos pontos bidimensionais obtidos da detecção das características faciais para estimar a orientação da face na imagem. O uso do POSIT apresenta como desvantagem o fato deste algoritmo ser aplicado a objetos rígidos, enquanto que a face é um objeto que possui grande variação de forma, não podendo ser descrita por um simples modelo rígido de características faciais. Por isto, o erro na estimação é alto, principalmente por utilizarmos um modelo tridimensional genérico da face, sem treinamento. Ainda assim, nosso sistema pode ser utilizado para interação, como por exemplo, identificação de gestos de cabeça, como sim ou não, etc.

Como trabalho futuro, sugerimos buscar um meio de estender o POSIT, permitindo sua aplicação em objetos deformáveis com resultados mais precisos. Chang *et al.* [CT02] já mostraram uma extensão deste algoritmo, possibilitando o seu uso na determinação de orientação de faces considerando a variação de expressões faciais do indivíduo. Os seus experimentos mostraram-se robustos e com alta porcentagem de convergência, provando sua viabilidade na estimação precisa da orientação de faces.

Referências Bibliográficas

- [ATAD02] E. Acosta, L. Torres, A. Albiol, and E. Delp. An automatic face detection and recognition system for video indexing applications. *International Conference on Acoustics, Speech, and Signal Processing*, 2002. 10, 16
- [Bar05] T. Barcelos. Interfaces prestativas baseadas em visão computacional e informação de contexto. Master's thesis, IME-USP, 2005. 3, 28
- [BEP96] S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head tracking. *Proceedings of the International Conference on Pattern Recognition (ICPR '96)*, 3(7276):611, 1996.
- [BG94] M. Buckland and F. Gey. The relationship between recall and precision. *Journal of the American Society for Information Science*, 45(1):12–19, 1994. 61
- [BH86] K. Berthold and P. Horn. Robot vision. *The MIT Press*, 1986. 7
- [BSSM05] R. Barbosa, F. Silva, T. Santos, and C. Morimoto. An extensible automatic video browsing tool. *XVIII Brazilian Symposium on Computer Graphics and Image Processing*, 2005. 57

- [BY95] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. *In Proc. 5th Intl. Conf. on Computer Vision, IEEE*, pages 374–381, 1995.
- [Cab05] M. Cabral. Interface baseada em gestos para ambientes de realidade virtual usando visão computacional. Master’s thesis, IME-USP, 2005. 3
- [Can86] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. 31
- [CC04] D. Cristinacce and T. Cootes. A comparison of shape constrained facial feature detectors. *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 375–380, 2004.
- [CET98] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *In Proc. 5th European Conference on Computer Vision*, 1998.
- [CM97] D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. *International Conference on Computer Vision and Pattern Recognition*, pages 750–755, 1997. 10, 14, 16, 51
- [CT02] C. Chang and W. Tsai. Determination of head pose and facial expression from a single perspective view by successive scaled ortographic approximations. *International Journal of Computer Vision*, 46(3):179–199, 2002. 62, 80
- [CT04] T. Cootes and C. Taylor. Statistical models of appearance for computer vision. *Imaging Science and Biomedical Engineering*, 2004. 13, 43, 48
- [DD95] D. DeMenthon and L. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, 1995. iv, 62, 64, 66

- [DH72] R. Duda and P. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972. 36
- [DMP96] T. Darrell, B. Moghaddam, and A. Pentland. Active face tracking and pose estimation in an interactive room, June 1996.
- [Enca] Wikipedia The Free Encyclopedia. Bresenham’s line algorithm. 38
- [Encb] Wikipedia The Free Encyclopedia. Real-time. 50
- [ETC98] G. Edwards, C. Taylor, and T. Cootes. Face recognition using the active appearance model. *5th European Conference on Computer Vision*, 2:581–695, 1998. 27
- [Fer01] R. S. Feris. Rastreamento eficiente de faces em um subespaço wavelet. Master’s thesis, IME-USP, 2001.
- [FGTK02] R. S. Feris, J. Gemmell, K. Toyama, and V. Krueger. Facial feature detection using a hierarchical wavelet face database. Technical Report MSR-TR-2002-05, Microsoft Research, 2002.
- [Fit01] P. Fitzpatrick. Head pose estimation without manual initialization. *Term Paper for MIT Course 6.892*, 2001.
- [FK00] B. Froba and C. Kublbeck. Orientation template matching for face localization in complex visual scenes. *International Conference on Image Processing*, pages 251–254, 2000.
- [FS95] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995. 12

- [FS00] L. Fan and K. Sung. Model-based varying pose face detection and facial feature registration in video images. *Proceedings of the eighth ACM international conference on Multimedia*, pages 295–302, 2000.
- [Gav99] D. Gavrilu. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.
- [GHC04] N. Gouriur, D. Hall, and J. L. Crowley. Estimating face orientation from robust detection of salient facial features. *Proceedings of Pointing, ICPR, International Workshop on Visual Observation of Deictic Gestures*, 2004. 27, 68
- [GHI99] J. L. Gabbard, D. Hix, and J. E. Swan II. User-centered design and evaluation of virtual environments. *Invited Paper to IEEE Computer Graphics and Applications*, 19(6):51–59, 1999.
- [GMB04] R. Gross, I. Matthews, and S. Baker. Constructing and fitting active appearance models with occlusion. *Proceedings of the IEEE Workshop on Face Processing in Video*, June 2004.
- [GW02] R. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, second edition, 2002. 29
- [HAMJ02] R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain. Face detection in color images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(5):696–706, May 2002. 10, 16
- [HL01] E. Hjelmas and B. K. Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83(3):236–274, 2001.

- [HZ98] J. Heinzmann and A. Zelinsky. 3-D facial pose and gaze point estimation using a robust real-time tracking paradigm. *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 142–147, 1998.
- [IBMK04] T. Ishikawa, S. Baker, I. Matthews, and T. Kanade. Passive driver gaze tracking with active appearance models. Technical Report CMU-RI-TR-04-08, Robotics Institute, Carnegie Mellon University, February 2004.
- [Jos99] S. H. Joseph. Optimal pose estimation in two and three dimensions. *Computer Vision and Image Understanding*, 73(2):215–231, 1999.
- [Kap] A. Kapoor. Automatic facial action analysis.
- [KBS00] V. Krueger, S. Bruns, and G. Sommer. Efficient head pose estimation with gabor wavelet networks. In *Proc. British Machine Vision Conference*, Bristol, UK, 2000.
- [Kin03] A. King. A survey of methods for face detection. Technical Report 992 550 627, University of Toronto, 2003.
- [KM96] R. Kothari and J. Mitchell. Detection of eye locations in unconstrained visual images. In *Proc. International Conference on Image Processing*, 1:519–522, September 1996. 27, 34, 37
- [Kru01] V. Krueger. Gabor wavelet networks for object representation. Technical Report 4245, University of Maryland, April 2001.
- [LK81] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981. 14
- [MB03] I. Matthews and S. Baker. Active appearance models revisited. Technical Report CMU-RI-TR-03-02, Robotics Institute, Carnegie Mellon University, April 2003.

- [MF00] C. Morimoto and M. Flickner. Real-time multiple face detection using active illumination. *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000. 10, 11, 27
- [OPB97] N. Oliver, A. Pentland, and F. Berard. Lafter: Lips and face real time tracker. *In Proc. Computer Vision and Patt. Recog.*, 1997.
- [PBM04] B. Pera, R. A. Barbosa, and C. H. Morimoto. Análise comparativa de diferentes espaços cromáticos para detecção de cor de pele. *VII Symposium on Virtual Reality*, 2004. 18, 78
- [RY92] D. Reisfeld and Y. Yeshurun. Robust detection of facial features by generalized symmetry. *11th IAPR International Conference on Pattern Recognition*, pages 117–120, 1992. 27
- [SAG99] M. Storrang, H. G. Andersen, and Erik Granum. Skin colour detection under changing lighting conditions. *7th Symposium on Intelligent Robotics Systems*, pages 187–195, jul 1999. 10, 16
- [San04] T. Santos. Segmentação automática de tomadas em vídeo. Master’s thesis, IME-USP, 2004. 59
- [SKK04] M. Savvides, B. Kumar, and P. Khosla. Eigenphases vs eigenfaces. *Proceedings of the 17th International Conference on Pattern Recognition*, pages 810–813, 2004. 79
- [STTK01] N. Streitz, P. Tandler, C. Tomfelde, and S. Konomi. Roomware: Towards the next generation of human-computer interaction based on an integrated design of real and virtual worlds. In J. Carroll, editor, *Human-Computer Interaction in the New Millenium*, pages 553–578. Addison-Wesley, 2001.

- [Sut65] I. Sutherland. The ultimate display, May 1965.
- [SYW96] R. Stiefelhagen, J. Yang, and A. Waibel. A model-based gaze tracking system. In *Proceedings of IEEE International Joint Symposia on Intelligence and Systems*, pages 304–310, 1996.
- [TA00] J. Terrillon and S. Akamatsu. Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images. *International Conf on Face and Gesture Recognition*, pages 54–61, 2000. 10, 17
- [TP94] M. Turk and A. Pentland. Eigenfaces for recognition. *Cognitive Neuroscience*, 3:71–86, 1994. 13, 21, 25
- [VD00] S. Valente and J. Dugelay. Face tracking and realistic animations for telecommunicant clones. *IEEE MultiMedia*, 7(1), 2000.
- [Vez02] V. Vezhnevets. Face and facial feature tracking for natural human-computer interface, 2002. 10, 16, 27
- [VJ01] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:511–518, 2001. 12, 27
- [WADP97] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfnder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.
- [WB95] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical Report TR95-041, University of North Carolina at Chapel Hill, 1995. 14

- [WY02] W. Widjojo and K. Yow. A color and feature-based approach to human face detection. *Seventh International Conference on Control, Automation, Robotics and Vision*, pages 508–513, 2002.
- [XBMK04] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2004.
- [YC97] K. Yow and R. Cipolla. Feature-based human face detection. *Image and Vision Computing*, 15:713–735, 1997. 11, 27
- [YKA02] M. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1), jan 2002. 8, 9
- [YLW98] J. Yang, W. Lu, and A. Waibel. Skin-color modeling and adaptation. In *ACCV (2)*, pages 687–694, 1998. 10, 16, 77
- [YS02] A. Yilmaz and M. Shah. Automatic feature detection and pose recovery for faces. *ACCV*, 2002. 27
- [YW95] J. Yang and A. Waibel. Tracking human faces in real-time. Technical Report CMU-CS-95-210, Carnegie Mellon University, nov 1995. 10, 16
- [ZCPR03] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003. 8
- [ZH01] Z. Zivkovic and F. Heijden. A stabilized adaptive appearance changes model for 3d head tracking. *2nd IEEE International Workshop on Recognition, Analysis*

and Tracking of Faces and Gestures in Real-time Systems (in connection with ICCV2001), page 175, July 2001.

- [ZJB05] Y. Zana, R. Cesar Jr, and R. Barbosa. Automatic face recognition system based on local Fourier-Bessel feature. *XVIII Brazilian Symposium on Computer Graphics and Image Processing*, 2005.
- [ZMI99] S. Zhai, C. Morimoto, and S. Ihde. Manual and gaze input cascaded (magic) pointing. *Proc. ACM SIGCHI - Human Factors in Computing Systems Conference*, pages 246–253, 1999. 3