IDENTIFICAÇÃO DE PADRÕES FACIAIS USANDO REDES NEURAIS ARTIFICIAIS

Adriano Martins Moutinho

Universidade Federal do Rio de Janeiro Núcleo de Computação Eletrônica

> Antonio Carlos Gay Thomé Ph. D.

Rio de Janeiro

2005

M934 Moutinho, Adriano Martins.

Identificação de Padrões Faciais Usando Redes Neurais Artificiais / Adriano Martins Moutinho– Rio de Janeiro, 2005.

214f.: il.;

Dissertação (Mestrado em Informática) - Universidade Federal do Rio de Janeiro, Instituto de Matemática, Núcleo de Computação Eletrônica, 2005.

Orientador: Antonio Carlos Gay Thomé

1. Reconhecimento de Padrões - Teses. 2. Reconhecimento de pele. - Teses 4. Redes Neurais. - Teses. 5. Detecção de faces - Teses. I. Antonio Carlos Gay Thomé (Orient.). II. Universidade Federal do Rio de Janeiro. Instituto de Matemática. Núcleo de Computação Eletrônica. III. Título

CDD

IDENTIFICAÇÃO DE PADRÕES FACIAIS USANDO REDES NEURAIS ARTIFICIAIS

Adriano Martins Moutinho

Dissertação submetida ao corpo docente do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro - UFRJ, como parte dos requisitos necessários à obtenção do grau de Mestre.

Aprovado por:

Prof.: _______Antonio Carlos Gay Thomé Ph. D. - Orientador

Prof.: _____

Luiz Biondi Neto D. Sc.

Prof.: _____

Manuel Lois Anido Ph. D.

Prof.: _____

Pedro Henrique Gouvea Coelho Ph. D.

Universidade Federal do Rio de Janeiro Núcleo de Computação Eletrônica

2005

Agradecimentos:

A possibilidade de haver vida em um planeta de algum sistema solar é bastante pequena. É necessário que seja um local quente, mas não demais; deve possuir atmosfera, mas não muito densa; deve possuir água, oxigênio, carbono e outros tantos requisitos.

E vida inteligente? Ainda mais difícil. Necessita de bilhões de anos de lenta evolução. Considerar que a vida na terra tenha sido guiada apenas por coincidências é, no mínimo, uma impossibilidade matemática. Acreditar que tudo aconteceu por acaso é achar que na vida se ganha muitas vezes na loteria.

Por que não pensar na hipótese de que há algo a modificar as chances e tornar isto tudo possível? Uma força, talvez divina, que mexe a balança a nosso favor, fazendo o improvável tornar-se viável, o inesperado tornar-se previsível, o impossível tornar-se nosso dia a dia.

Minha fé é que, assim como a criação do mundo onde vivemos, a conclusão desta dissertação não teria sido possível sem a ajuda de Deus, sem que Ele tivesse agrupado todos os átomos que formam o papel em que este trabalho foi impresso, ou todos os elétrons que circulam ordenadamente pelos computadores por onde tudo isto passou.

Portanto, o meu primeiro agradecimento vai para Deus, pois só poderia ter sido Ele quem colocou no mundo criaturas tão boas e perseverantes quanto os meus Pais, Antonio e Marilena, a quem agradeço também, pois se não insistissem MUITO na minha capacidade nada disso teria sido possível.

Para a minha esposa Cristiane devo o maior de todos os agradecimentos, pois mesmo com o incentivo dos meus pais minha vida seria a de um engenheiro frustrado, que não foi capaz de encontrar sozinho a vocação para a docência, descoberta por influência de minha amada e linda professora e companheira.

Para o meu orientador, professor Thomé, também guardo um agradecimento especial, pois acreditou em mim desde o primeiro momento, bem como todo o grupo de pesquisadores e amigos do NCE, organização que também me permitiu um ótimo suporte financeiro para uma época conturbada da minha vida.

Devo também agradecimentos ao professor Biondi da UERJ, que despertou em mim interesse pelos sistemas inteligentes, e ao meu amigo de videogame Sérgio Telles, pois sem suas opiniões loucamente abrangentes tudo teria sido mais difícil.

E finalmente para todos os meus muitos amigos, alunos e professores. Não pensem que isto tudo é uma coincidência, se Ele nos colocou aqui todos juntos é para um propósito maior, para que a vida continue...

RESUMO

O reconhecimento automático de faces é uma das vertentes do universo dos métodos biométricos, que se propõe a processar e reconhecer a identidade de uma pessoa a partir de imagens capturadas de sua face. O reconhecimento biométrico de faces é hoje uma aplicação de grande interesse para a sociedade, mas que ainda apresenta inúmeros desafios em aberto tanto para a pesquisa como para uma implementação comercial.

Um dos problemas comumente encontrados nesta área é o de separar a região da face, caracterizada por conter olhos, nariz e boca, do fundo e de outros objetos possivelmente presentes na imagem capturada. Localizar a área de uma imagem que corresponde a uma face, também chamado detecção de faces, é parte integrante de qualquer sistema automático de reconhecimento de identidade, e constitui um problema bastante complexo sendo, inclusive, estudado separadamente.

Para tornar a localização de faces computacionalmente possível, foram empregados, nesta dissertação, métodos de processamento de imagens tradicionais, em conjunto com novos métodos desenvolvidos como a correção de iluminação por aproximação linear, a detecção de cor de pele e indícios de face, e a concepção de um extrator de características faciais usando uma versão ampliada da projeção poligonal, adaptada para ser usada em imagens em tons de cinza.

Nesta dissertação descreve-se a pesquisa realizada sobre métodos biométricos relacionados à face, bem como a concepção de um conjunto de procedimentos e algoritmos a serem empregados no tratamento de imagens coloridas e na localização das regiões onde há alta probabilidade de conter uma face. Obteve-se, durante o desenvolvimento de um protótipo em MATLAB, desempenho superior aos trabalhos apresentados na literatura da área, principalmente em relação ao número de testes e comparações necessárias para efetuar a localização da face em uma imagem colorida.

ABSTRACT

Face recognition is an application that belongs to the biometric family and its major purpose is to recognize somebody's identity by processing images containing faces. Biometric face recognition is becoming a very important application for security issues, however it still presents some open challenges to researchers in order to develop a commercial implementation.

One of the common problems concerning face recognition is to segment the face region, characterized by eyes, nose and mouth, from the background and from other objects possibly present in the source image. Segmentation of human faces, also called face detection, is a mandatory part of any automatic face identity recognition system, and it constitutes a very complex problem, normally studied separately.

In order to make face detection computationally possible, traditional image processing techniques were used and new methods were developed, such as illumination correction by linear approximation, skin and face clue detection, and the conception of a face characteristics extraction by using a improved version of polygonal projection, adapted to be applied on a grayscale image.

This dissertation describes a research about biometric systems and the development of procedures and algorithms for face detection. It also presents a prototype implementation of a face detector, which is able to find the region where human faces are likely to be. The prototype, developed in MATLAB, achieves a superior performance compared to others research projects in this area of expertise, mainly because it takes less comparisons and tests to find face localization in a color image.

Lista de figuras

1.1	Definição de face, menor retângulo, ou próximo, que contem olhos,	ર
12	Diagrama em blocos simplificado do sistema detector de faces	3
2.1	Exemplo de íris humana	7
2.2	Exemplo de íris localizadas pelo método dos dois círculos (esquerda) e com modelos para as pálpebras superiores e inferiores (esquerda)	7
2.3	Mudança de coordenadas efetuada para compensar variações de escala das iris.	8
2.4	Exemplo de Impressão digital adquirida com tinta sobre papel	9
2.5	Tratamento utilizado para encontrar minúcias em impressões digitais. As linhas papilares estão representadas em preto. (MARQUES, 2004)	11
2.6	Exemplos de minúcias freqüentemente encontradas em impressões	
	digitais (MARQUES, 2004).	11
2.7	Exemplos encontrados em quatro diferentes trabalhos sobre pontos	
	fiduciais, extraídos para reconhecimento de faces. (HUMANSCAN,	
	2005) na esquerda superior, (ARCA e CAMPADELLI et al., 2003)	
	na parte superior direita, (JESORSKY e KIRCHBERG et al., 2001)	
	na esquerda inferior e (LIAO e LI, 2000) na parte direita inferior. $$.	15
2.8	reconstruções de uma mesma face com 1, 8, 16, 24, 32, 40 e 48	
	variáveis mais importantes da análise de PCA (autofaces) (DANNER	
	e DATTA, 1999)	17
2.9	Reconstrução de 5 faces feita com apenas 50 variáveis (DANNER e	
	DATTA, 1999)	18
4.1	Exemplo de extração de bordas, a busca pelos olhos torna-se mais	
	fácil pois o espaço de busca foi diminuído.	34
4.2	Exemplo de detecção de faces usando características colorimétricas,	
	além de uma análise de formas faciais elípticas (KRUPPAAND e	
	BAUER et al., 2002). (Em cores) \ldots \ldots \ldots \ldots \ldots \ldots \ldots	35
4.3	Diversas medidas antropométricas da face	36
4.4	As razões $\frac{X_{boca}}{Y_{boca}} = \frac{X_{olho}}{Y_{olho}}$ são diferentes, o que permite separar os olhos	
	da boca	37
4.5	Dezesseis particularidades marcadas para cálculo dos Jets	39

5.1	Modelo de neurônio artificial	43
5.2	Perceptron, apenas uma camada de neurônios	44
5.3	Modelo MLP	44
5.4	Gráficos das funções de ativação mais usadas em redes MLP, log-	
	sigmóide à esquerda e tan-sigmóide à direita. (Em cores) $\ldots \ldots$	47
5.5	Exemplo de uso do banco de dados de validação para parada anteci-	
	pada no ponto de melhor generalização. (Em cores)	49
5.6	Distribuição criada artificialmente para possuir correlação. Em ver-	
	melho são plotados os autovetores q_N da matriz de covariância (PCA),	
	onde $\lambda_1 \in \lambda_2$ indicam os autovalores associados. (Em cores)	50
5.7	Distribuição da figura 5.6 rotacionada de forma que os eixos sejam os	
	autovetores da matriz de covariância (PCA). (Em cores)	51
5.8	Exemplo da evolução do algoritmo K-means com $K = 6$ clusters,	
	aplicado em uma distribuição criada artificialmente. (Em cores) $\ . \ .$	53
5.9	Reprodução por cruzamento. Um ponto de cruzamento é escolhido	
	aleatoriamente e dois filhos são criados $\ldots \ldots \ldots \ldots \ldots \ldots \ldots$	55
5.10	Exemplo de figura em tons de cinza, a matriz numérica mostra os	
	valores de luminância da pequena região selecionada	57
5.11	Exemplo de figura colorida no espaço de cores RGB, as matrizes	
	numéricas mostram os valores de vermelho, verde e azul da pequena	
	região selecionada. (Em cores)	58
5.12	Exemplo de misturas aditivas - vermelho, verde e azul são as cores	
	primárias. (Em cores)	59
5.13	Espaço de cores XYZ representado pelas variáveis X e Y. A parte em	
	colorida representa o espaço $RGB,$ enquanto a parte cinza representa	
	o espaço não representável em RGB . (Em cores)	60
5.14	Exemplo de gradientes de cores no espaço LAB , os valores de $a \in b$	
	representam parcelas de croma e os valores de L representam brilho.	
	$(Em \ cores) \ldots \ldots$	61
5.15	Exemplo de cálculo de histograma com 256 "bins" em uma figura com	
	256 níveis de cinza	63

5.16	Exemplo de equalização de histograma, obtendo-se histograma cu-	
	mulativo aproximadamente linear. A imagem original está na figura	
	5.15	64
5.17	Exemplo de aplicação do filtro de maioria	65
5.18	Exemplo de operação de fechamento efetuada por uma operação de	
	dilatação seguida de uma erosão	66
5.19	Exemplo de segmentação efetuada pelo algoritmo de componentes	
	conexos. (Em cores)	67
5.20	Máscara para convolução que resulta no cálculo do operador Sobel	
	(esquerda) e do laplaciano (direita) de uma imagem	68
5.21	Máscara para convolução que resulta no cálculo do operador Prewitt	
	(esquerda) e Roberts (direita)	69
5.22	Exemplo de detecção de bordas por operador Sobel, Prewitt e Canny	70
6.1	Três exemplos de extração de pontos de pele através da técnica de	
	componentes conexos, aplicada dentro de um limiar de 10 pontos dos	
	valores RGB. O círculo na extremidade da linha azul indica o ponto	
	inicial selecionado pelo usuário. (Em cores)	73
6.2	Configuração da rede neural usada para detectar pele. Não está rep-	
	resentado o Bias. (Em cores)	74
6.3	Distribuição dos pontos de pele do banco de dados. (Em cores)	75
6.4	Distribuição dos pontos de pele separada em três gráficos que possuem	
	os eixos $Vermelho \times Verde$, $Vermelho \times Azul$ e $Verde \times Azul$. (Em	
	cores)	75
6.5	Distribuição dos pontos de não-pele . (Em cores)	76
6.6	Distribuição dos pontos de não-pele e pele mostrados no mesmo gráfico.	
	Pontos de pele são mostrados em vermelho, pontos de não-pele são	
	mostrados em vermelho e os pontos em comum aos dois bancos de	
	dados são mostrados em magenta. (Em cores) $\ \ldots \ \ldots \ \ldots \ \ldots \ \ldots$	77
6.7	Testes com o sistema detector de pele, imagens coloridas do banco de	
	dados FERET e as preto e branco mostram a saída do detector, onde	
	preto representa pele e branco não-pele. (Em cores)	80

6.8	Testes com o sistema detector de pele onde é aplicado previamente	
	uma equalização de histograma na componente de iluminação. Preto	
	representa pele e branco não-pele. (Em cores)	81
6.9	Detector de pele aplicado em figura contendo outros objetos além de	
	faces. Preto representa pele e branco não-pele. (Em cores) \ldots .	82
6.10	Detector de pele aplicado em figura contendo outros objetos além de	
	faces. Preto representa pele e branco não-pele. (Em cores) \ldots .	82
6.11	Filtro de maioria (direita) aplicado depois do detector de pele (esquerda)	83
6.12	Operações de fechamento, preenchimento e remoção de áreas menores	
	que 5000 pixels	84
6.13	Exemplos de detecção de bordas na diminuição do espaço de busca.	
	As bordas detectadas estão representadas em preto. (Em cores) $\ .$.	86
6.14	Exemplos de extração de pontos de borda-pele, representados em	
	preto nas quatro figuras em teste. Os pontos de borda pele são ex-	
	traídos através de uma operação ${\rm E}$ entre os pontos de pele e os pontos	
	de bordas. (Em cores) \ldots	87
6.15	Exemplo de extração de pontos de borda-pele, representados em preto	
	na figura em teste. (Em cores) \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots	88
6.16	Exemplo de extração de pontos de borda-pele em uma figura com	
	mais de uma face. Os pontos de borda-pele estão representados em	
	preto. (Em cores)	89
6.17	Histograma Altura/Largura de 1935 faces do banco de dados FERET	89
6.18	Exemplo de seleção de pontos de borda-pele no eixo dos X. (Em cores)	90
6.19	Exemplo de seleção de pontos de borda-pele no eixo dos Y. (Em cores)	91
6.20	Exemplo de uma distribuição de pontos criada artificialmente $\ \ . \ . \ .$	91
6.21	Exemplos do algoritmo de seleção de pontos aplicado em uma dis-	
	tribuição criada artificialmente, com intervalos iguais a 4, 3, 2 e 1 $$	
	pixel	92
6.22	Exemplo do algoritmo de redução de pontos aplicado nos pontos de	
	borda-pele clusterizados (número de clusters igual a 3). As cores	
	vermelho, azul e roxo indicam os clusters, cujo centro está indicado	
	por um círculo na imagem inferior esquerda. O algoritmo de redução	
	é executado em cada cluster de forma independente. (Em cores) $\ . \ .$	93

6.23	Exemplos contidos no banco de dados de faces (ROWLEY e KANADE, 1999) 95
6.24	Exemplos contidos no banco de dados de não-faces (ROWLEY e
	KANADE, 1999)
6.256.26	Sistema de correção de iluminação
	calculado
6.27	Quatro exemplos de equalização de histograma, com as mesmas ima-
	gens de face da figura 6.26
6.28	Mascaramento dos pontos laterais
6.29	Mascaramento dos pontos laterais
6.30	Diagrama da rede neural usada para detectar faces com mapa de bits.
	Não está representado os Bias
6.31	Exemplos de faces do banco de dados de teste
6.32	Exemplo de projeção poligonal usando um quadrado como base $\ .\ .\ .\ 107$
6.33	Exemplos de distâncias medidas a partir da aresta esquerda do quadrado.
	O conjunto de medidas extraídas é (2, 3, 3, 3, 8, 3, 5, 6, 4, 3, 2). (Em
0.04	cores)
6.34	Exemplos de distâncias medidas a partir da aresta inferior do quadrado
	O conjunto de medidas extraídas é $(5, 2, 3, 4, 3, 5, 5, 6, 7, 4, 2, 5)$.
	(Em cores)
6.35	Exemplos de como os olhos, nariz e boca são mais escuros que o resto
	da face, justificando que a imagem seja invertida antes de aplicar a
	projeção poligonal
6.36	Projeção poligonal feita do centro para fora da figura, com o obje-
	tivo de registrar nas distâncias as características faciais que estejam
	eventualmente presentes no centro das imagens
6.37	exemplo do primeiro conjunto (fora para dentro) de projeção poligonal
	de uma face
6.38	exemplo do segundo conjunto (centro para fora) de projeção poligonal
	de uma face
6.39	exemplo do primeiro conjunto (fora para dentro) de projeção poligonal
	de uma face levemente rotacionada

6.4	10 exemplo do segundo conjunto (centro para fora) de projeção poligonal
	de uma face levemente rotacionada $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 113$
6.4	11 exemplo do segundo conjunto (centro para fora) de projeção poligonal
	de um exemplo de não-face
6.4	42~exemplo do segundo conjunto (centro para fora) de projeção poligonal
	de um exemplo de não-face
6.4	13 Diagrama ampliado do sistema de detecção de faces
6.4	4 Exemplo de espelhamento vertical de uma imagem. As faces, devido
	a simetria, não perderão suas características típicas
6.4	15 Exemplo do ajuste fino de redução por espelhamento. Todas as de-
	tecções à esquerda sofreram espelhamento e foram novamente verifi-
	cadas pela RNA. As detecções em vermelho não foram consideradas
	faces com o espelhamento, sendo removidas após o ajuste fino. \ldots . 120
6.4	6 Exemplo do ajuste fino de redução por centróide. Todos os quadrados
	amarelos são "vizinhos de centróide", tendo sido reduzidos a apenas
	um calculado através da média aritmética ponderada em relação ao
	índice de confiança. O quadrado em vermelho não foi incluído na
	média pois não possui "vizinhos de centróide". (Em cores) \ldots . 121
6.	17 Exemplo do ajuste por recomparações sucessivas. A marcação em
	vermelho representa a entrada do algoritmo e a marcação em amarelo
	a face melhor ajustada. (Em cores)
7.	Protótipo do sistema automático de detecção de faces desenvolvido
	em MATLAB (MATHWORKS, 2005). São aplicados em seqüência,
	da esquerda para a direita; em cima: Filtro de pele, filtro de maioria,
	operação de fechamento, preenchimento e remoção de pequenas áreas,
	detector de bordas, localizador de borda-pele, redução dos pontos de
	borda-pele; em baixo: faces detectadas, redução por espelhamento,
	redução por centróide e recomparação sucessiva
7.	2 Falso-positivos em vermelho, detectados devido ao aumento da faixa
	de tamanho de procura pela face. Verifica-se que a redução por es-
	pelhamento eliminou um dos falso-positivos. (Em cores)

7.3	Distribuição do tamanho da face no banco de dados FERET. Cada
	face é representada por um "+" na posição que representa a sua
	largura (em x) e a sua altura (em y)
7.4	Sistema de estimativa do tamanho da face. A partir da saída do filtro
	de preenchimento e fechamento, escolhe-se a maior área presente na
	figura e considera-se a face. A raiz quadrada da área será considerada
	o lado da face. (Em cores) \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 127
7.5	Exemplos de erro na estimativa do tamanho da face, onde a maior
	área conectada de pele, mostrada em azul, é considerada como face.
	Na parte superior a pouca iluminação gerou uma estimativa pequena
	demais. Na parte inferior o fundo com cor semelhante à pele gerou
	uma estimativa grande demais. (Em cores)
7.6	Tamanho de busca em pixels por $\%$ de acertos do algoritmo genético
	usado para ajustar as variáveis da estimativa do tamanho da face $~$. $~$.130
7.7	Exemplos de imagens presentes no banco de dados FERET. (Em cores) 132
7.8	Exemplos de imagens presentes no banco de dados FERET, com faces
	marcadas manualmente em azul. (Em cores) \hdots
7.9	Exemplos do cálculo de <i>acerto</i> e <i>amais</i> em imagens do banco de
	dados FERET testadas pelo sistema. O cálculo das variáveis é feito
	comparando a marcação automática feita pelo sistema (em amarelo)
	e manual (em azul). O número na parte superior da detecção correta
	representa o índice de confiança, como descrito na seção 6.4.1.2. (Em
	cores)
7.10	$8~{\rm exemplos}$ de detecções corretas feitas pelo sistema do teste $8~{\rm da}$
	tabela 7.2. O número acima do retângulo que demarca a detecção
	representa o índice de confiança atribuído pela rede neural, conforme
	exposto na seção 6.4.1.2. (Em cores) $\ldots \ldots \ldots \ldots \ldots \ldots \ldots 137$
7.11	$8~{\rm exemplos}$ de detecções incorretas feitas pelo sistema do teste $8~{\rm da}$
	tabela 7.2. O número acima do retângulo que demarca a detecção
	representa o índice de confiança atribuído pela rede neural, conforme
	exposto na seção 6.4.1.2. (Em cores) \ldots \ldots \ldots \ldots \ldots \ldots 138
7.12	Exemplos do acompanhamento da posição da face com o uso do ajuste
	de recomparação sucessiva. (Em cores) \hdots

7.13	Teste mostrando a resistência do sistema à rotação da face. (Em cores) 140
7.14	Exemplos de acompanhamento da posição da face com várias mu-
	danças de expressão. (Em cores)
7.15	Exemplos do acompanhamento da posição da face onde ocorreu um
	erro de detecção. (Em cores)
7.16	Exemplo de detecção de múltiplas faces. Seleção feminina de hande-
	bol no pan-americano de 2003 (disponível em http://www.photoegrafia.com.br/).
	A maioria das faces foi encontrada. Um falso-positivo totalmente in-
	correto, na primeira fila, quarta jogadora. Detecção efetuada entre
	20 e 30 × 30 pixels de tamanho. (Em cores) $\dots \dots \dots$
7.17	Exemplo de detecção de múltiplas faces. Time do Flamengo do Rio de
	Janeiro de 2001 (disponível em http://www.cmrevolution.com.br/).
	Iluminação forte demais. A maioria das faces foi encontrada. Dois
	falso-positivos completamente incorretos, nas pernas de dois jogadores.
	Detecção efetuada entre 20 e 30×30 pixels de tamanho. (Em cores) . 144
7.18	Exemplo de detecção de múltiplas faces. Time do Flamengo do Rio de
	Janeiro campeão Carioca de 2004 (disponível em http://lancenet.ig.com.br/).
	Grande complexidade e presença de várias faces, algumas não frontais.
	Quatro falso-positivos completamente incorretos. Detecção efetuada
	entre 20 e 30 \times 30 pixels de tamanho. (Em cores) \ldots \ldots \ldots \ldots 145
7.19	Exemplo de detecção de múltiplas faces com oclusão facial. Foto
	da Formatura 2003.2 do Curso de Matemática da UESC (Universi-
	dade Estadual de Santa Cruz) (disponível em http://www.uesc.br/).
	Perdeu-se uma face por problemas de iluminação além das que estão
	parcialmente ocultas. Detecção efetuada entre 20 e 30×30 pixels de
	tamanho. (Em cores) $\ldots \ldots 146$
7.20	Exemplo de detecção de múltiplas faces. Adriano e seu pai Antônio.
	Nesta foto foi possível aplicar o algoritmo de redução de pontos. Um
	falso-positivo devido à cor de pele da camisa. Foi aplicada a redução
	de pontos, o número de clusters foi definido em 10, e as faces são
	procuradas des de 35 até 70 \times 70 pixels. (Em cores) \ldots \ldots \ldots . 147

Exemplo de detecção de múltiplas faces. Apenas dois falso-positivos			
encontrados e uma face não foi detectada. Foi aplicada a redução			
de pontos, o número de clusters foi definido em 10, e as faces são			
procuradas desde 35 até 70 \times 70 pixels. (Em cores) $~\ldots~\ldots~\ldots~\ldots~148$			
Exemplo de detecção efetuada por (MOGHADDAM e PENTLAND,			
1997) (tracejado), que possui um critério menos restritivo, e da de-			
tecção efetuada por esta dissertação (linha contínua amarela). (Em			
cores) $\ldots \ldots 150$			
Dois exemplos de detecção correta de faces segundo (TAKÁCS e			
WECHSLER, 1995). A figura completa representa a detecção do			
trabalho e em amarelo a detecção proposta por esta dissertação. (Em			
cores)			

Lista de tabelas

6.1	Exemplos de pele e não-pele. (Em cores)
6.2	Resultados de 10 redes para detecção de pele. Acertos (%) no subcon-
	junto diminuído representa o desempenho em um banco de dados que
	possui apenas exemplos que se repetiram mais de 5 vezes. "Acertos
	(%)nos 39 × 106" pontos representa o desempenho em todo o banco
	de dados
6.3	Comparação deste trabalho com outros métodos de detec ção de pele . $\ 79$
6.4	Configurações de 32 redes para detecção de faces usando mapa de
	bits. Os resultados do treinamento estão na tabela 6.5 $\hfill \ldots \hfill 105$
6.5	Resultados do treinamento de 32 redes para detecção de faces usando
	mapa de bits. As colunas "acertos em faces" e "acertos em não-faces"
	representam a porcentagem de acerto no banco de dados de testes
	contendo faces e não-faces respectivamente. A configuração destas
	redes está na tabela 6.4
6.6	Configurações de 32 redes neurais para detecção de faces com projeção
	poligonal. Os resultados estão na tabela 6.7
6.7	Resultados do treinamento de 32 redes para detecção de faces usando
	projeção poligonal. As colunas "acertos em faces" e "acertos em não-
	faces" representam a porcentagem de acerto no banco de dados de
	testes contendo faces e não-faces respectivamente. A configuração
	destas redes está na tabela 6.6
6.8	tabela comparativa entre a extração de características por projeção
	poligonal e por mapa de bits
7.1	Exemplos do cálculo de T_{INI} e T_{FIM} , Faixa de procura pela definida
	pela estimativa do tamanho da face
7.2	Configuração dos diversos parâmetros do sistema e avaliação dos
	testes executados no banco de dados FERET
8.1	Comparação dos métodos Usando o banco de dados FERET 149

Lista de Termos e Abreviações

Íris	Pequeno músculo circular colorido que envolve a pu- pila.	6
Afinamento	Operação morfológica aplicada em imagens binárias que resulta em linhas com espessura igual a um pixel.	12
Antropometria	Estudo das medidas do corpo humano para uso em classificação e comparação.	33
Autofaces	Método de reconhecimento de faces que faz uso da análise de componentes principais (PCA).	16
Binarização	Transformação de uma imagem cujo objetivo é ter um valor binário para cada pixel.	12
Borda-pele	Pontos que possuem cor de pele e também represen- tam bordas.	85
Busca Gulosa	Busca de maximização ou minimização onde a cada passo o melhor resultado é tomado como base para o progresso do algoritmo.	122
Canny	Operador para extração de bordas.	69
CCD	Charge Coupled Device - tem a função de conver- ter fótons em elétrons sendo usado em câmeras fo- tográficas ou vídeo.	56
DARPA	Agência de Defesa e Pesquisa em Projetos Avançados dos Estados Unidos da América.	131
Deteção de Pele	Método computacional que se propõe a detectar as regiões que correspondem à pele humana.	24
DNA	Ácido Desoxirribonucléico.	21
DTW	Dynamic Time Warping.	19

DVD-ROM	Digital Versatile Disc - Disco óptico com capacidade em torno de 4.7Gb (ou 9.4Gb) usado para guardar arquivos.	132
Energia de Projeção	Valor inicial de uma projeção poligonal, a ser di- minuído pelo valor da luminância na direção de projeção.	109
Esclerótica	Membrana fibrosa e branca que envolve a iris.	7
Espelhamento	Operação que inverte o eixo dos y em uma imagem, obtendo efeito semelhante ao de um espelho.	118
Face	Figura que representa uma face humana, menor retângulo que contém os olhos, nariz e boca.	94
FERET	Facial Recognition Technology: banco de dados usado para reconhecimento e detecção de faces.	79
Filtro oval	Mascaramento de pontos laterais em uma imagem.	100
Frame	Figura capturada por câmera de vídeo.	35
FTIR	Frustrated Total Internal Reflection - Sistema de cap- tura de impressão digital.	10
Gaussiana	Função de densidade de probabilidade de Gauss, também conhecida como distribuição normal.	28
Histograma	Gráfico de contagem do número de repetições de com- portamento de uma variável.	26
Holístico	Relativo a holismo; que busca um entendimento in- tegral dos fenômenos; Quando um procedimento de reconhecimento de faces é dito holístico o reconheci- mento é feito diretamente sobre a imagem.	16
ICA	Independent Components Analysis - Análise de com- ponentes independentes.	18

Identificação	Sistema biométrico de identificação - Quando um sis- tema utiliza-se de um método biométrico para encon- trar a identidade de um indivíduo desconhecido em um banco de dados.	6
Imagens Binárias	Imagens que possuem apenas dois valores possíveis para cada pixel, ligado (branco) ou desligado (preto).	12
Jet	Conjunto de coeficientes extraídos com o filtro de ga- bor que contém características de particularidades da face.	40
K-means	Algoritmo de clusterização.	92
LAB	Codificação usada para imagens coloridas onde L contém a componente de iluminação enquanto a e b contém componentes de croma.	61
Laplacian	Operador para extração de bordas.	33
Linhas papilares	Linhas que formam as impressões digitais.	10
Mapa de bits	Matriz contendo um certo número de bits para cada ponto de uma imagem.	95
Minúcias	Detalhes encontrados em impressões digitas.	10
MLP	Percéptron de múltiplas camadas.	30
Motores de busca	Servidores que oferecem serviço de busca de in- formações pela Internet.	2
Não-face	Figura que não representa uma face humana.	94
Papiloscopista	Especialista em linhas papilares de impressões digi- tais.	10
PCA	Análise de componentes principais.	102
PCA	Principal Components Analysis ou análise de compo- nentes principais.	16

PDF	Função de densidade de probabilidade.	28
Pixel	Menor parte ou ponto de uma matriz de imagem.	97
Pontos fiduciais	Pontos importantes encontrados para reconhecimento de faces.	14
Projeção poligonal	Método de extração de características utilizado pri- mariamente para reconhecimento de caracteres.	107
Pupila	Orifício circular escuro no centro da íris, responsável pela adaptação do indivíduo à diferença de iluminação ambiente.	7
RGB	Codificação usada em imagens coloridas onde cada ponto é representado pelos valores de vermelho,verde e Azul.	24
RNA	Rede Neural Artificial.	38
Scanner	Equipamento responsável pela captura de imagens estáticas provenientes de impressões.	10
Segmentar	Segmentar objetos em imagens - Encontrar e separar uma sub-região de uma imagem contendo um objeto ou informação.	8
Sobel	Operador para extração de bordas.	33
SOM	Rede neural baseada em um mapa auto-organizável.	29
SSE	Summed Squared Errors - Somatório dos erros eleva- dos ao quadrado.	18
Verificação	Sistema biométrico de verificação - Quando um sistema utiliza-se de um método biométrico para certificar-se da identidade que um determinado in- divíduo diz possuir.	6
Vizinhos de D4	Vizinhança de pontos considerando quatro direções possíveis excluindo as diagonais; apenas as direções horizontais e verticais.	120

Vizinhos de D8	Vizinhança de pontos considerando as oito direções possíveis; incluindo as diagonais.	120
WebCam	Câmera de vídeo de baixa resolução normalmente em- pregada em vídeo conferência e comunicações de vídeo na internet.	138
YRB	Codificação usada em imagens coloridas onde cada ponto é representado por um valor representando a luminância (Y) e dois representando a crominância (R-Y e B-Y).	28

Atenção! Este trabalho possui figuras em cores...

Esta dissertação possui figuras cujo conteúdo deverá ser visualizado em cores. Entretanto, devido aos custos ainda proibitivos de acesso à cópias e impressões coloridas, o leitor poderá deparar-se aqui com uma versão proveniente de uma impressora ou copiadora laser, onde as figuras estarão em escala de cinzas.

Neste caso, as barras de cores desta folha estarão impressas de acordo com seus respectivos níveis de cinza, e o leitor é encorajado a obter a versão digital em formato PDF com o conteúdo deste trabalho, disponível na biblioteca ou na Internet, no endereço eletrônico http://www.adrianomoutinho.com.

Para simplificar o acompanhamento do texto, as figuras cujo conteúdo necessitar ser avaliado em cores serão marcadas, na sua legenda, por uma indicação entre parênteses "(em cores)". Neste caso, a figura e possíveis referências no texto serão melhor compreendidas se o leitor que obtiver uma cópia em preto e branco visualizar esta figura no arquivo PDF em um computador que possua o software Adobe Acrobat Reader, que pode ser obtido em http://www.adobe.com.

Peço sinceras desculpas por este inconveniente, esperando que este pequeno eventual obstáculo não impeça o leitor de avaliar o trabalho...

Sumário

1	Inti	rodução	1
	1.1	Motivação	1
	1.2	Proposta	2
2	Mé	todos biométricos	5
	2.1	Reconhecimento de íris	6
	2.2	Reconhecimento de impressões digitais	9
	2.3	Reconhecimento de faces	13
		2.3.1 Métodos baseados em medidas de características faciais	14
		2.3.2 Métodos baseados na informação da imagem	16
	2.4	Reconhecimento de assinaturas e padrões de escrita	19
	2.5	Reconhecimento de voz	20
	2.6	Outros reconhecimentos biométricos	21
	2.7	Conclusões	22
3	Det	tecção de Pele - Revisão Bibliográfica	24
	3.1	Técnicas ponto a ponto	24
		3.1.1 Técnicas usando regras e limiares	24
		3.1.2 Técnicas estatísticas não paramétricas	26
		3.1.3 Técnicas estatísticas paramétricas	27
		3.1.4 Técnicas usando redes neurais artificiais	29
	3.2	Uso de informações de vizinhança em métodos de detecção de pele	30
	3.3	Conclusões	31
4	Det	tecção de Faces - Revisão Bibliográfica	32
	4.1	Motivação e aplicações	32
	4.2	Métodos baseados em características faciais	33
		4.2.1 Detecção por bordas	33
		4.2.2 Detecção por características colorimétricas	34
		4.2.3 Detecção usando movimento	35
		4.2.4 Detecção por modelos faciais	36
	4.3	Métodos baseados na imagem	37
		4.3.1 Métodos de detecção facial usando redes neurais	37

		4.3.2	Métod	os de detecção facial usando grafos	38	
	4.4	conclu	sões .		41	
5	Fun	damer	tação	teórica	42	
	5.1	Redes	neurais	artificiais	42	
		5.1.1	O mod	lelo MLP	42	
		5.1.2	Valida	ção e parada antecipada	47	
		5.1.3	Uso de	e Componentes Principais (PCA) em Redes Neurais	48	
	5.2	Métod	o de clu	ısterização K-means	51	
	5.3	3 Algoritmos genéticos				
	5.4	Técnie	rocessamento de imagens	55		
		5.4.1	Repres	sentação matricial de uma imagem e espaços de cores	56	
		5.4.	L.1 E	spaço de cores RGB	57	
		5.4.	L.2 E	spaço de cores YC_RC_B	58	
		5.4.	L.3 E	spaço de cores CIE XYZ	59	
		5.4.	L.4 E	spaço de cores CIE LAB	61	
		5.4.2	Equali	zação de Histograma	62	
		5.4.3	Técnic	as de morfológicas de processamento de imagem	64	
		5.4.	8.1 F	iltro de maioria	64	
		5.4.	8.2 O	peração de fechamento	65	
		5.4.4	Segme	ntação usando componentes conexos	66	
		5.4.5	Métod	os de extração de bordas de uma imagem	68	
6	Cor	ntribui	ções de	esta dissertação	71	
	6.1	Sistem	a locali	zador de indícios de face	71	
		6.1.1	Sistem	a localizador de tons de pele	71	
		6.1.	.1 0	btenção do banco de dados de tons de pele	72	
		6.1.	l.2 U	so de rede neurais no localizador de tons de pele	73	
		6.1.2	Sistem	a localizador de bordas	84	
		6.1.2	2.1 M	otivação para uso em detecção de faces	85	
		6.1.3	Obten	ção dos indícios de face - pontos de borda-pele	85	
	6.2	Segme	ntador	de sub-imagens	86	
		6.2.1	Métod	o de redução do número de pontos de borda-pele	89	
	6.3	Detec	ão de f	aces com redes neurais	94	

		6.3.1 Ba	anco de dados de faces e não-faces	94
		6.3.2 M	étodos de pré-processamento das imagens do banco de dados	
		de	e faces e não-faces	95
		6.3.2.1	Correção de iluminação por aproximação linear	96
		6.3.2.2	Equalização de histograma	99
		6.3.2.3	Filtro oval	99
		6.3.3 O	rdem de aplicação dos métodos de pré-processamento $\ . \ . \ .$	101
		6.3.4 M	éto dos de extração de características e codificação da face $\ .$	102
		6.3.4.1	Aplicação de Componentes principais (PCA) $\ . \ . \ .$.	102
		6.3.4.2	Mapa de bits	103
		6.3.4.3	Projeção poligonal de figuras em níveis de cinza $\ .\ .\ .$	106
	6.4	Esquema	final do Sistema localizador de faces $\ . \ . \ . \ . \ . \ .$	115
		6.4.1 Aj	justes finos no detector facial	116
		6.4.1.1	Redução por espelhamento	118
		6.4.1.2	Redução por centróide	119
		6.4.1.3	Ajuste por recomparações sucessivas	121
7	\mathbf{Des}	envolvim	ento e Avaliação do Protótipo	123
7	Des 7.1	senvolvim Protótipo	ento e Avaliação do Protótipo o do sistema detector de faces	123 123
7	Des 7.1	senvolvim Protótipo 7.1.1 Ac	ento e Avaliação do Protótipo o do sistema detector de faces	123 123 123
7	Des 7.1	envolvim Protótipo 7.1.1 Ac 7.1.1.1	ento e Avaliação do Protótipo o do sistema detector de faces	 123 123 123 125
7	Des 7.1 7.2	envolvim Protótipo 7.1.1 Ac 7.1.1.1 Resultado	ento e Avaliação do Protótipo o do sistema detector de faces	 123 123 123 125 131
7	Des 7.1 7.2	envolvim Protótipo 7.1.1 Ac 7.1.1.1 Resultado 7.2.1 Ca	ento e Avaliação do Protótipo o do sistema detector de faces daptações e ajustes ao banco de dados Obtendo uma estimativa do tamanho da face os usando o banco de dados FERET aracterísticas do banco de dados FERET	 123 123 125 131 131
7	Des 7.1 7.2	envolvim Protótipo 7.1.1 A 7.1.1.1 Resultado 7.2.1 Ca 7.2.2 Ci	ento e Avaliação do Protótipo o do sistema detector de faces daptações e ajustes ao banco de dados Obtendo uma estimativa do tamanho da face os usando o banco de dados FERET aracterísticas do banco de dados FERET ritérios automáticos de avaliação da posição da face	 123 123 125 131 131 132
7	Des 7.1 7.2	envolvim Protótipo 7.1.1 A 7.1.1.1 Resultado 7.2.1 Ca 7.2.2 Ci 7.2.3 A	ento e Avaliação do Protótipo o do sistema detector de faces daptações e ajustes ao banco de dados Obtendo uma estimativa do tamanho da face os usando o banco de dados FERET aracterísticas do banco de dados FERET ritérios automáticos de avaliação da posição da face valiação de desempenho usando o banco de dados FERET	 123 123 125 131 131 132 135
7	Des 7.1 7.2 7.3	envolvim Protótipo 7.1.1 Ac 7.1.1.1 Resultado 7.2.1 Ca 7.2.2 Cu 7.2.3 Av Resultado	ento e Avaliação do Protótipo o do sistema detector de faces daptações e ajustes ao banco de dados Obtendo uma estimativa do tamanho da face os usando o banco de dados FERET aracterísticas do banco de dados FERET ritérios automáticos de avaliação da posição da face valiação de desempenho usando o banco de dados FERET os usando uma WebCam	 123 123 125 131 131 132 135 138
7	Des 7.1 7.2 7.3 7.4	envolvim Protótipo 7.1.1 Ac 7.1.1.1 Resultado 7.2.1 Ca 7.2.2 Cu 7.2.3 Av Resultado Resultado	ento e Avaliação do Protótipo o do sistema detector de faces daptações e ajustes ao banco de dados Obtendo uma estimativa do tamanho da face os usando o banco de dados FERET aracterísticas do banco de dados FERET citérios automáticos de avaliação da posição da face valiação de desempenho usando o banco de dados FERET os usando uma WebCam os usando imagens com mais de uma face	 123 123 125 131 132 135 138 141
7	Des 7.1 7.2 7.3 7.4 Cor	Protótipo 7.1.1 A 7.1.1 A 7.1.1.1 Resultado 7.2.1 Ca 7.2.2 Ca 7.2.3 A Resultado Resultado Resultado	ento e Avaliação do Protótipo o do sistema detector de faces daptações e ajustes ao banco de dados Obtendo uma estimativa do tamanho da face os usando o banco de dados FERET os usando o banco de dados FERET aracterísticas do banco de dados FERET oritérios automáticos de avaliação da posição da face os usando uma WebCam os usando imagens com mais de uma face	 123 123 125 131 132 135 138 141 149
7	Des 7.1 7.2 7.3 7.4 Cor 8.1	Protótipo 7.1.1 Ac 7.1.1 Ac 7.1.1.1 Resultado 7.2.1 Ca 7.2.2 Cu 7.2.3 Ac Resultado Resultado nclusões F Conclusõe	ento e Avaliação do Protótipo o do sistema detector de faces daptações e ajustes ao banco de dados Obtendo uma estimativa do tamanho da face os usando o banco de dados FERET os usando o banco de dados FERET aracterísticas do banco de dados FERET os usando de dados FERET aracterísticas do banco de dados FERET os usando de dados FERET os usando de dados FERET os usando de dados ference os usando de dados ference os usando de dados ference os usando uma WebCam os usando imagens com mais de uma face os usando imagens com mais de uma face	 123 123 125 131 132 135 138 141 149
7	Des 7.1 7.2 7.3 7.4 Cor 8.1 8.2	Protótipo 7.1.1 Ac 7.1.1 Ac 7.1.1 Resultado 7.2.1 Ca 7.2.2 Cu 7.2.3 Ac Resultado Resultado nclusões F Conclusõe	ento e Avaliação do Protótipo o do sistema detector de faces	 123 123 125 131 132 135 138 141 149 149 152
7	Des 7.1 7.2 7.3 7.4 Cor 8.1 8.2 8.3	envolvim Protótipo 7.1.1 A 7.1.1 A Resultado 7.2.1 Ca 7.2.2 Cu 7.2.3 A Resultado Resultado nclusões F Conclusõe Conclusõe	ento e Avaliação do Protótipo o do sistema detector de faces daptações e ajustes ao banco de dados Obtendo uma estimativa do tamanho da face os usando o banco de dados FERET aracterísticas do banco de dados FERET aracterísticas do banco de dados FERET citérios automáticos de avaliação da posição da face os usando uma WebCam os usando imagens com mais de uma face os sobre o sistema de localização de faces o sobre as contribuições desta dissertação ações finais e trabalhos futuros	 123 123 125 131 132 135 138 141 149 149 152 153
7	Des 7.1 7.2 7.3 7.4 Cor 8.1 8.2 8.3 8.4	Senvolvim Protótipo 7.1.1 A 7.1.1 A Resultado 7.2.1 Ca 7.2.2 Ca 7.2.2 Ca 7.2.3 A Resultado Resultado nclusões F Conclusõe Conclusõe Considera Trabalhos	ento e Avaliação do Protótipo o do sistema detector de faces daptações e ajustes ao banco de dados Obtendo uma estimativa do tamanho da face os usando o banco de dados FERET os usando o banco de dados FERET aracterísticas do banco de dados FERET aritérios automáticos de avaliação da posição da face os usando uma WebCam os usando imagens com mais de uma face os sobre o sistema de localização de faces o sobre as contribuições desta dissertação ações finais e trabalhos futuros	 123 123 125 131 132 135 138 141 149 149 152 153 154

Apêndice de rotinas	do MATLAB
---------------------	-----------

Referências Bibliográficas

1 Introdução

Separar a região da face em uma fotografia contendo possivelmente outros objetos é uma tarefa computacionalmente complexa, sendo foco de estudo de diversos trabalhos como (WISKOTT e FELLOUS et al., 1997), (HERPERS e KATTNER et al., 1995), (GOVINDARAJU, 1996), (SENIOR e HSU et al., 2002) e (ROWLEY e KANADE, 1999), entre outros.

Embora várias destas publicações também enfoquem o reconhecimento da identidade da face do indivíduo em questão, existe uma diferença entre detecção e reconhecimento de faces. No caso da detecção de faces, o objetivo é encontrar, a princípio de forma automática, todas as faces presentes em uma figura, segmentando-as da imagem original.

Já o reconhecimento de faces exige que o sistema possa identificar o indivíduo baseando-se em um banco de dados previamente cadastrado, o que exige uma prévia localização da face em questão. Isto torna o foco deste trabalho, a detecção de faces, um importante módulo inicial de um sistema capaz de reconhecer indivíduos em imagens.

Esta dissertação está organizada em oito capítulos, sendo esta introdução o primeiro deles. O capítulo 2 é um estudo do estado da arte dos métodos biométricos de reconhecimento de identidade, mostrando as vantagens e desvantagens de cada um deles, incluindo o reconhecimento de faces. Os capítulos 3 e 4 são uma revisão bibliográfica dos métodos de detecção de pele e faces respectivamente, que foram tomados como base no desenvolvimento do sistema de detecção de faces, descrito no capítulo 6. O capítulo 5 é uma fundamentação teórica das técnicas de processamento de imagens e redes neurais empregadas nesta dissertação, e finalmente nos capítulos 7 e 8 são mostrados respectivamente, os resultados e conclusões encontrados. Um apêndice ao final desta dissertação contém algumas das rotinas que implementam em MATLAB (MATHWORKS, 2005) o protótipo detector de faces.

1.1 Motivação

A vigilância em vídeo e aplicações de segurança são a motivação mais forte para o desenvolvimento de um sistema de detecção de faces, pois qualquer sistema de segurança de vídeo baseado em reconhecimento de identidade deverá segmentar primeiramente a região de interesse, ou seja, o rosto do indivíduo, sabendo-se que a figura poderá conter outros objetos além da face.

Uma outra aplicação do sistema de detecção de faces, além das relacionadas com segurança, é o catálogo de imagens, que poderá separar automaticamente figuras que contém pessoas, criando um índice para facilitar a procura. Motores de busca na Internet como (GOOGLE, 2005) e (YAHOO!, 2005) possuem busca por imagens, mas não empregam nenhum método de detecção de objetos ou faces.

Técnicas de compressão, principalmente para vídeo conferência, também podem fazer uso de sistemas de detecção de faces, o que possibilita manter uma melhor qualidade nas regiões onde rostos foram encontrados. A expressão facial durante a comunicação é normalmente a mais importante informação presente no vídeo ou na imagem, mas pode não ser o detalhe mais fino encontrado. Sistemas de compressão integrados à detecção de faces poderão disponibilizar uma maior banda de transmissão para partes da imagem que são realmente relevantes à comunicação.

1.2 Proposta

Propõe-se um método computacional que receba uma fotografia colorida e localize a posição de todos os rostos humanos que lá estejam presentes. Neste trabalho, para efeito prático, será considerado como face o menor retângulo que corta de uma fotografia os olhos, nariz e boca de um indivíduo, como mostra o retângulo amarelo menor na figura 1.1. Esta definição não deve ser demasiadamente restrita, pois como mostra a figura 1.1, retângulos um pouco maiores também representarão faces. A seção 7 mostrará um critério mais adequado para testes automáticos do sistema detector de faces.

Para realizar a detecção de faces, um sistema automático foi idealizado, como mostra o diagrama em blocos da figura 1.2. Inicialmente um sistema detector de indícios de face, descrito na seção 6.1, indica quais as partes da imagem original possuem maior probabilidade de conter faces, através de informações de pele e bordas. Esta informação é então passada ao segmentador de sub-imagens descrito na seção 6.2, que cortará a figura original em várias sub-imagens com tamanho mínimo e máximo definido pelo usuário, que serão analisadas por um sistema neural de detecção de faces.



Figura 1.1: Definição de face, menor retângulo, ou próximo, que contem olhos, nariz e boca



Figura 1.2: Diagrama em blocos simplificado do sistema detector de faces

O sistema neural de detecção de faces, descrito na seção 6.3, foi treinado para detectar se uma pequena sub-parte da figura original contém uma face frontal, o que faz com que a localização proposta nesta dissertação seja restrita a faces frontais. Contudo, testes no sistema de localização de faces descrito na seção 7 mostraram que algumas leves rotações ou inclinações da face são permitidas, sem grandes prejuízos à eficiência do localizador.

UFRJ/IM/NCE

Ainda no diagrama da figura 1.2, o módulo de ajustes finos é necessário para evitar múltiplas detecções e validar as faces detectadas, como descrito na seção 6.4.1. Ao final do processo, todas as faces presentes na figura original que forem detectadas serão delimitadas por retângulos, seguindo a definição da figura 1.1.

2 Métodos biométricos

Sistema ou método biométrico é aquele capaz de identificar os padrões freqüentemente apresentados pelos seres humanos, e tem por objetivo reconhecer a identidade ou a presença de uma pessoa.

Vários tipos de padrões biométricos são usados como forma de identificar pessoas, sendo inclusive empregados como provas em processos criminais. Com o advento da computação, foi possível projetar sistemas automáticos mais confiáveis e rápidos, e que permitem comparar padrões biométricos em bancos de dados com grande número de indivíduos.

Os padrões biométricos mais comumente encontrados na literatura são: impressões digitais, detalhes da face ou da íris, assinatura, padrões de voz, detalhes na arcada dentária, tamanho e geometria das mãos, entre outros. Segundo (JAIN e PANKANTI et al., 1999), quaisquer padrões biométricos devem seguir sete critérios para serem válidos como métodos de reconhecimento de identidade:

- Universalidade A maioria dos indivíduos devem possuir o padrão indicado.
- Exclusividade O padrão deve ser único para cada indivíduo.
- Permanência O padrão deve permanecer igual mesmo com o passar do tempo.
- Facilidade de coleta O padrão deverá ser facilmente coletável, ou seja, sua obtenção deverá possuir o menor número de complicadores possível.
- Resistência a fraudes O sistema baseado em um padrão biométrico deve ser o mais resistente possível a fraudes.
- Desempenho Os padrões devem ser, preferencialmente, comparados de forma computacionalmente rápida.
- Aceitação Refere-se à receptividade que o padrão biométrico possui frente ao público.

Os sistemas biométricos podem ser empregados para verificação ou identificação. Nos sistemas de verificação, a identidade de um usuário é informada pelo mesmo e atestada por sua biometria para, por exemplo, permitir a sua entrada em um determinado setor ou dar acesso aos arquivos de um computador. Isso faz com que estes sistemas sejam quase sempre cooperativos, ou seja, o usuário auxilia o sistema colocando o seu dedo para recolher a impressão digital ou ficando perfeitamente alinhado frente a uma câmera para captar a sua íris ou face.

Em sistemas de identificação, a identidade de uma pessoa é totalmente desconhecida e deseja-se comparar características biométricas desta com as de um banco de dados, obtendo-se com isto, a identidade. Estes sistemas não devem se basear nas premissas de cooperatividade, pois não há como garantir o alinhamento ou a qualidade de imagem em qualquer caso onde se deseja fazer a identificação. Em certos ambientes talvez apenas imagens de faces não frontais ou pedaços incompletos de impressão digital estejam presentes, o que torna o desenvolvimento de um sistema automático um desafio ainda mais difícil.

2.1 Reconhecimento de íris

A íris é um pequeno músculo circular colorido que envolve a pupila, mostrado na figura 2.1, e que contém padrões que são diferentes de um indivíduo para o outro (ADLER, 1965). Um dos aspectos fundamentais no reconhecimento de íris é que os padrões biométricos formados permanecem semelhantes desde o estágio embrionário onde são formados até a idade adulta do indivíduo (KRONFELD, 1968), sendo ainda mais similares se comparados após a infância. Isso faz com que os padrões de íris sejam mais confiáveis dos que os padrões de face ou de voz, além de serem mais significativos do que a impressão digital (WILDES, 1997).

Embora padrões de íris já tenham sido utilizados para verificação visual desde o século XIX (BERTILLON, 1885), os primeiros sistemas automáticos de reconhecimento de identidade por padrões presentes de íris foram reportados por (FLOM e SAFIR, 1987), (JOHNSON, 1991), (DAUGMAN, 1993) e (WILDES, 1997).

Um dos obstáculos do reconhecimento de íris é obter uma imagem com resolução e nível de detalhes suficientes para possibilitar o reconhecimento, sem que haja necessidade de dispositivos específicos como apoiadores de queixo ou olhos. O trabalho em (WILDES e ASMUTH et al., 1996) propõe que uma câmera a 20cm de distância dos olhos do indivíduo, usando uma lente de 80mm, obtenha uma imagem contendo 256 pixels. Já em (DAUGMAN, 1993), uma câmera é colocada a 15cm com uma lente de 330mm, obtendo de 100 a 200 pixels de informação. Em



Figura 2.1: Exemplo de íris humana

ambos os sistemas existe um cuidado para que a iluminação não seja baixa demais, sob o risco de se obter uma imagem sem contraste, ou alta demais, o que poderia causar incômodo à pessoa em teste. Alguns sistemas podem evitar o incômodo da iluminação (WILDES, 1997) fazendo a aquisição da imagem em infravermelho.

Mesmo com a cooperação do indivíduo, não se pode esperar que apenas a íris esteja presente na imagem. Desta forma, os sistemas automáticos de reconhecimento devem lidar com a localização da íris na imagem como o primeiro módulo do sistema. Em (DAUGMAN, 1993), é proposto um sistema baseado em busca de bordas que possui um modelo formado por dois círculos, um maior para a borda entre a esclerótica e a íris, e outro menor para a borda entre a pupila e a íris. Já em (WILDES e ASMUTH et al., 1996), a busca por bordas é também baseada em modelos para as pálpebras superiores e inferiores.



Figura 2.2: Exemplo de íris localizadas pelo método dos dois círculos (esquerda) e com modelos para as pálpebras superiores e inferiores (esquerda)

Depois de segmentar a íris da imagem original, uma mudança de coordenadas é feita com o intuito de tornar o sistema invariante quanto a possíveis variações de tamanho das íris encontradas, como mostra a figura 2.3.



Figura 2.3: Mudança de coordenadas efetuada para compensar variações de escala das iris.

Após a mudança de variáveis e normalização entre 0 e 1 das variáveis $r \in \theta$, é feita uma codificação de características da íris. Em (DAUGMAN, 1993), esta codificação é feita através de uma transformada bidimensional de Gabor, seguida de uma quantização de fase como mostrado na equação 2.1. Onde $I(\rho, \phi)$ é a imagem da íris, $r_0 \in \theta_0$ são as coordenadas do local onde os coeficientes estão sendo calculados. As constantes $\alpha \in \beta$ representam as escalas e ω a freqüência do filtro de Gabor.

$$h_{\{\text{Re,Im}\}} = \text{sgn}_{\{\text{Re,Im}\}} \int_{\phi} \int_{\rho} I(\rho, \phi) e^{-i\omega(\theta_0 - \phi)} e^{\frac{-(r_0 - \rho)^2}{\alpha^2}} e^{\frac{-(\theta_0 - \phi)^2}{\beta^2}} \rho d\rho d\phi \qquad (2.1)$$

A quantização da equação 2.1 é executada por $\text{sgn}_{\{\text{Re},\text{Im}\}}$ (função sinal), o que faz com que o valor de $h_{\{\text{Re},\text{Im}\}}$ seja 0 ou 1 (um bit) para a parte real e 0 ou 1 (um bit) para a parte imaginária.

Em (DAUGMAN, 1993), 2048 bits (256 bytes) extraídos da equação 2.1 codificam uma íris, o que possibilita que a identificação de uma pessoa seja feita através da comparação deste vetor de bits com outros no banco de dados. Uma possibilidade é que a comparação seja feita através da distância de Hamming, definida na equação 2.2, onde A e B são os vetores codificados das íris que se deseja comparar.

$$HD = \frac{1}{N} \sum_{j=1}^{N} A_J \otimes B_J \tag{2.2}$$

Se A e B representam a mesma íris, o valor de HD na equação 2.2 será pequeno. Este método permite que o sistema automático busque em um banco de dados uma íris de identidade conhecida que possua um valor mínimo de HD entre outra íris de identidade desconhecida. Este valor mínimo representa a íris do banco de dados que mais se assemelha àquela desconhecida, o que possivelmente indica que ambas possuem a mesma identidade.

2.2 Reconhecimento de impressões digitais

Impressões digitais são pequenos sulcos presentes principalmente na superfície dos dedos humanos. A sua formação, assim como a íris, dá-se no período embrionário e os padrões presentes são únicos para cada indivíduo (MALTONI e MAIO et al., 1991). A figura 2.4 mostra um exemplo de impressão digital.



Figura 2.4: Exemplo de Impressão digital adquirida com tinta sobre papel.

O estudo de impressões digitais é reportado desde o século XVII, mas foi no fim do século XIX que surgiram os primeiros sistemas de catálogo, como o sistema Henry, que possibilita a comparação visual de impressões digitais (HENRY, 1900). No sistema Henry as impressões eram indexadas e catalogadas por especialistas de acordo com a presença de espirais, arcos e voltas, que facilitavam a procura de uma impressão desconhecida em um banco de dados ainda em papel.

A partir de 1960, houve um grande interesse acerca de sistemas automáticos de reconhecimento de identidade a partir de impressões digitais. Estes sistemas são normalmente formados por quatro etapas distintas: a aquisição da imagem e localização da impressão, o pré-processamento, a extração de características, e finalmente a comparação das características obtidas com um banco de dados de impressões digitais.

A aquisição da imagem de uma impressão digital pode ser feita através da marcação com tinta em um papel, onde a pessoa é normalmente orientada por um profissional treinado. Esta impressão em papel é então convertida para imagem digital por um *scanner*. Este método de aquisição pode gerar imagens de baixa qualidade devido a variações na pressão feita no papel e ruído na conversão para digital, e ainda tornar necessário um sistema automático como o proposto em (MARQUES, 2004), capaz de separar a impressão digital presente no papel de manchas, anotações e marcações.

A aquisição de impressões digitais ficou mais acessível com o advento de técnicas como o FTIR *(Frustrated Total Internal Reflection)* onde o dedo é apoiado sobre uma superfície de vidro e iluminado por um LED, que sensibiliza um CCD de acordo com os sulcos e linhas da impressão digital (HARTMAN, 1996).

Na comparação visual de impressões digitais, o especialista, também chamado papiloscopista, procura por descontinuidades nas linhas da impressão digital, as linhas papilares. Estas descontinuidades são referidas na literatura como minúcias (LEE e GAENSSLEN, 1991), e sua correta localização permite a comparação e identificação de uma impressão digital. Na figura 2.5 são mostrados dois exemplos de minúcias freqüentemente encontradas em impressões digitais, como o término abrupto ou bifurcação de uma linha papilar. Dessa forma, o trabalho do especialista é encontrar e marcar a posição e o tipo das minúcias encontradas.

Outra característica também empregada na comparação entre impressões digitais é a localização dos poros presente na impressão. No entanto, devido à dificuldade de obtenção de imagens contendo informações de poros, o uso das minúcias, como mostrado na figura 2.5 é mais comum.
Minúcia relativa ao término Minúcia relativa à bifurcação de uma linha papilar de uma linha papilar

Figura 2.5: Tratamento utilizado para encontrar minúcias em impressões digitais. As linhas papilares estão representadas em preto. (MARQUES, 2004)

O pré-processamento de imagens envolvido em impressões digitais requer um método que possibilite a extração de minúcias, o que normalmente é executado com a binarização, que faz com que cada ponto da imagem seja preto ou branco, seguido do afinamento (GONZALEZ e WOODS, 1992), que faz com que todas as linhas da imagem da impressão digital tenham largura de 1 pixel, simplificando a localização das minúcias. A figura 2.6 mostra o pré-processamento de uma impressão digital, o que é considerado um método clássico (MARQUES, 2004).



Figura 2.6: Exemplos de minúcias freqüentemente encontradas em impressões digitais (MARQUES, 2004).

Após a binarização e o afinamento, as minúcias podem ser encontradas de forma trivial, bastando para isso varrer toda a imagem a procura de pontos pretos que contenham apenas um vizinho, o que representaria um ponto de término da linha papilar, ou de ponto pretos que possuam três vizinhos, o que representaria uma bifurcação, como mostra a figura 2.5. Quando uma minúcia é encontrada as suas coordenadas x e y são catalogadas, bem como o ângulo θ que esta faz com a linha papilar. Quanto mais minúcias são encontradas melhor será a representação da impressão digital.

No entanto, a binarização seguido de afinamento pode gerar ruídos e levar a detecção de falsas minúcias, que são causadas principalmente pela baixa qualidade da imagem original. Dessa forma, alguns trabalhos propõem variações ao método clássico como o uso de redes neurais para detectar minúcias a partir de um pré-processamento usando filtros de Gabor em (LEUNG e LAU et al., 1991), ou a remoção de minúcias próximas, consideradas falsas em (NETO, 2000). Em (MARQUES, 2004) é também proposta uma versão neural do pré-processamento baseado em um filtro de Gabor, que possui menor custo computacional.

De posse da localização e tipo das minúcias, uma comparação é efetuada para identificar a impressão digital, usando para isto um prévio cadastro destas minúcias em um banco de dados. Em (MALTONI e MAIO et al., 1991), é proposto um modelo de comparação que computa quantas minúcias em comum duas impressões digitais possuem. Devido a variações na extração, pode haver uma translação ou rotação entre as minúcias da impressão digital cadastrada e a que se deseja verificar, além disso, uma impressão pode ter menos minúcias que a outra. Para efetuar a comparação, (MALTONI e MAIO et al., 1991) propõe que cada minúcia da impressão a ser verificada seja alinhada com outra do banco de dados usando-se uma transformação linear.

Após esse alinhamento, as minúcias serão comparadas usando a equação 2.3, onde (x_a, y_a) e (x_b, y_b) são as coordenadas das minúcias $a \in b$, e $\theta_a \in \theta_b$ os ângulos.

$$\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \le r_0$$

min $(|\theta_a - \theta_b|, 360 - |\theta_a - \theta_b|) \le \theta_0$ (2.3)

Assim, de acordo com a equação 2.3, uma minúcia a será considerada correspondente a b caso sejam verdadeiras as equações, definidos os valores de limiar θ_0 e r_0 . Em (MALTONI e MAIO et al., 1991), comparações entre minúcias cadastradas manualmente permitiram propor empiricamente valores de θ_0 e r_0 iguais a 22,5° e 15 pixels. Considerando todos os alinhamentos de translação e rotação possíveis entre as minúcias de a e b, o maior número de correspondências encontradas é considerado a impressão digital presente no banco de dados mais semelhante àquela cuja identidade se deseja encontrar.

O método proposto por (MALTONI e MAIO et al., 1991) permite obter uma estimativa para a probabilidade de que duas impressões digitais diferentes obtenham a correspondência de minúcias, ou seja, sejam consideradas erroneamente como provenientes de uma mesma pessoa, o que é dependente do número de minúcias das duas impressões digitais. Em um caso criminal, a impressão recolhida de um objeto, conhecida como impressão latente, não possui muito mais do que uma dezena de minúcias, sendo comparada com uma recolhida em ambiente controlado, podendo conter em torno de 50 minúcias.

Em (PANKANTI e PRABHAKAR et al., 2002), é demonstrado que a probabilidade de uma impressão digital contendo apenas 12 minúcias seja falsamente considerada de mesma identidade em relação à outra contendo 36 minúcias é de 6, 10×10^{-8} . Já em (MARQUES, 2004) calcula-se que a probabilidade de falsa correspondência entre duas impressões digitais com 12 e 60 minúcias é de 1,914 × 10^{-12} , o que já pode ser considerada baixa o suficiente para constituir prova quase irrefutável em um processo criminal.

2.3 Reconhecimento de faces

O reconhecimento de faces é um dos mais antigos interesses computacionais, e assim como a impressão digital acredita-se que cada face possui um conjunto de características que a faz única, exclusiva.

Os trabalhos envolvendo reconhecimento de faces podem ser classificados em dois grupos: os métodos baseados em medidas de características faciais e os métodos baseados na informação da imagem propriamente dita.

2.3.1 Métodos baseados em medidas de características faciais

Os métodos baseados em medidas receberam interesse principalmente forense, sendo (GALTON, 1888) um dos primeiros trabalhos na área. Reconhecer faces com base em medidas exige que sejam marcados, manual ou automaticamente, pontos representando características da face como os olhos, boca, nariz, queixo e outros.

Nos trabalhos baseados em medidas, as faces normalmente estão em posição frontal como em (CRAW e ELIS et al., 1987) e (KANADE, 1973). No entanto, em (IP e NG, 1994), 12 medidas são tomadas manualmente sobre fotografias de faces em posição frontal e perfil, sendo comparadas usando a teoria de evidência de Dempter e Shafer (DEMPSTER, 1968) (SHAFER, 1976).

Sistemas automáticos baseados em medidas necessitam que sejam implementados detectores capazes de encontrar os pontos que representam características da face, também conhecidos como pontos fiduciais. Muitos trabalhos se especializaram apenas em encontrar estes pontos, não tendo a intenção inicial de propriamente reconhecer a identidade da face. Em (GARCIA e SIM et al., 2001), os pontos fiduciais são encontrados através da análise dos dados de decomposição wavelet, enquanto em (CRAW e TOCK et al., 1992) os pontos são encontrados a partir da comparação de modelos poligonais criados para o contorno da face, dos olhos, da boca e outros.

Os sistemas mais promissores de reconhecimento de face por medidas são baseados nos coeficientes complexos extraídos da transformada de Gabor (WISKOTT e FELLOUS et al., 1996) e (WISKOTT e FELLOUS et al., 1997), também conhecidos como Jets. Nestes trabalhos, os pontos fiduciais são encontrados através do cálculo da similaridade entre coeficientes extraídos da transformada e os obtidos em um banco de dados. A transformada de Gabor também é útil na detecção de faces, sendo descrita em maiores detalhes na seção 4.3.2.

A figura 2.7 mostra alguns exemplos de pontos fiduciais. À esquerda superior, um exemplo de marcação feita manualmente encontrada no banco de dados BioID (HUMANSCAN, 2005). Na esquerda inferior os pontos fiduciais são encontrados através da comparação entre um detector de bordas e modelos propostos por (JESORSKY e KIRCHBERG et al., 2001). Na parte direita, tanto superior (ARCA e CAMPADELLI et al., 2003) quanto inferior (LIAO e LI, 2000), exemplos de medidas extraídas usando transformada de Gabor, onde os pontos são encontrados através da comparação dos jets, descrito com mais detalhes na seção 4.3.2.



Figura 2.7: Exemplos encontrados em quatro diferentes trabalhos sobre pontos fiduciais, extraídos para reconhecimento de faces. (HUMANSCAN, 2005) na esquerda superior, (ARCA e CAMPADELLI et al., 2003) na parte superior direita, (JESORSKY e KIRCHBERG et al., 2001) na esquerda inferior e (LIAO e LI, 2000) na parte direita inferior.

Com os pontos fiduciais marcados, a comparação entre os vetores que o representam pode ser feita de várias formas. Em (STAROVOITOV e SAMAL, 2000), é usado a distância euclidiana entre pontos marcados manualmente no indivíduo em teste para encontrar a face mais semelhante presente no banco de dados. Em (LIAO e LI, 2000) os pontos encontrados são calculados com uma variação do método do vizinho mais próximo. Nos trabalhos que usam a transformada de Gabor, a semelhança entre os Jets é muitas vezes usada como forma de comparação para o reconhecimento de faces (ARCA e CAMPADELLI et al., 2003), entretanto em (LIU e WECHSLER, 2002) é implementado um discriminante de Fisher para reconhecer a identidade a partir dos coeficientes da transformada de Gabor.

2.3.2 Métodos baseados na informação da imagem

Os métodos baseados em medidas, descritos na seção 2.3.1, propõem que uma face seja reconhecida através da avaliação das distâncias ou da forma com que se dispõem as características típicas da face, como os olhos, nariz, boca, queixo e outros. Entretanto, alguns métodos se baseiam apenas na informação da imagem, sem que haja necessidade de se localizar previamente os pontos fiduciais da face.

Os sistemas de reconhecimento de faces baseados na informação da imagem são também referidos como procedimentos holísticos (ZHAO e CHELLAPPA et al., 2004), ou seja, que efetuam o reconhecimento diretamente na imagem, sem dividir o processo em busca e medição de pontos na face.

A maioria dos métodos holísticos para reconhecimento de faces constitui uma variação ou aperfeiçoamento da técnica conhecida como Autofaces, que consiste na criação de vetores ortonormais através do procedimento de Karhunen-Loève, mais conhecido como análise de componentes principais (PCA). A técnica de PCA está descrita mais detalhadamente na seção 5.1.3 pois é também empregada nesta dissertação no processamento do vetor de dados para o treinamento das redes neurais da seção 6.1.1 e 6.3.

O uso de PCA para o reconhecimento de faces foi originalmente proposto por (SIROVICH e KIRBY, 1987), sendo mais tarde desenvolvido por (TURK e PEN-TLAND, 1989) e (TURK e PENTLAND, 1991), e finalmente por (PENTLAND e MOGHADDAM et al., 1994). O reconhecimento de identidade por autofaces propõe que um banco de dados de imagens de faces em tons de cinza, previamente processadas e alinhadas, tenham sua dimensão reduzida por PCA, sendo então possível caracterizar as faces do banco de dados com um número bem diminuído de variáveis.

O alinhamento das faces é bastante importante antes da aplicação da análise de componentes principais (TURK e PENTLAND, 1991), para assim permitir que esta técnica possa extrair as características mais comuns a todas as faces presentes nas variáveis mais relevantes do espaço projetado, ou seja, que possuem os autovalores mais altos (HAYKIN, 1994). Dessa forma, antes da aplicação de PCA, todas as faces presentes no banco de dados são cortadas para conter apenas os olhos, nariz,

boca e queixo, estando estes sempre em posições relativamente próximas em todas as imagens.

Usando os valores das 50 variáveis mais relevantes, segundo (TURK e PEN-TLAND, 1991), é possível reconstituir um rosto efetuando a operação inversa da projeção de PCA (HAYKIN, 1994). A quantidade de variáveis necessária varia de acordo com o número de faces presentes no banco de dados. Em (PENTLAND e MOGHADDAM et al., 1994) são usadas 100 variáveis enquanto em (DANNER e DATTA, 1999) apenas 50.

A figura 2.8 mostra reconstruções de uma mesma face com 1, 8, 16, 24, 32, 40 e 48 variáveis mais importantes encontradas na análise de PCA feita por (DANNER e DATTA, 1999). Nota-se que com apenas 24 variáveis já é possível uma boa aproximação do original, ficando muito próximo deste com 48 variáveis.



Figura 2.8: reconstruções de uma mesma face com 1, 8, 16, 24, 32, 40 e 48 variáveis mais importantes da análise de PCA (autofaces) (DANNER e DATTA, 1999)

A figura 2.9 também mostra reconstruções com apenas 50 variáveis, feitas por (DANNER e DATTA, 1999), de 5 faces originalmente com 200x300 pixels, ou seja uma redução de um espaço de dados de 60000 (200×300) valores de cinza, ou seja, 60000 variáveis, para apenas 50, ainda guardando boa parte das características presentes nas faces.



Faces reconstituídas com as 50 variáveis mais importantes do espaço projetado por PCA



O reconhecimento de faces é efetuado primeiramente projetando a figura de identidade desconhecida, corretamente alinhada, no espaço de dados de dimensão reduzida, o que é feito apenas multiplicando a matriz de transformação obtida na análise PCA pela imagem original (HAYKIN, 1994). Em uma segunda etapa, é calculado o somatório dos erros ao quadrado (SSE) entre o resultado da projeção, com apenas 50 variáveis em (TURK e PENTLAND, 1991), e todas as imagens do banco de dados, também projetadas em PCA. O menor valor de SSE corresponde à imagem presente no banco de dados mais semelhante àquela cuja identidade se pretende reconhecer.

Alguns trabalhos propuseram variações da técnica de autofaces, como o Fisherfaces, que emprega o discriminante linear de Fisher ao invés de PCA para efetuar a redução de dimensionalidade (BELHUMEUR e HESPANHA et al., 1997). Em (LADES e BARTLETT et al., 1998) foi empregado Análise de Componentes Independentes (ICA), enquanto em (YANG, 2002) foi utilizado uma versão não linear de PCA. Entretanto, a necessidade de um alinhamento preciso entre as faces do banco de dados diminuiu o número de trabalhos baseados em princípios semelhantes ao autofaces, bem como da maioria dos procedimentos holísticos.

2.4 Reconhecimento de assinaturas e padrões de escrita

O reconhecimento de assinaturas tem normalmente grande interesse jurídico, principalmente devido à alta aceitação de padrões de escrita como prova processual (ELLEN, 1956). Especialistas em grafologia normalmente comparam assinaturas e padrões de escrita baseados em padrões de proporção de ocupação das palavras, espaços e regularidades entre as características de escrita como curvatura e pressão feita sobre o papel (HARRISON, 1958) (EDSON J. R. JUSTINO e SABOURIN, 1998).

Os métodos automáticos de reconhecimento de assinatura podem ser baseados em padrões de escrita em tempo real ("online"), onde uma câmera ou aparato específico não apenas efetuará a captura da assinatura como também os aspectos temporais que podem vir a ser importantes para a identificação. Alguns destes dispositivos também permitem capturar a pressão e ângulo da caneta ao longo da assinatura, também usados como padrões biométricos (TAPPERT e WAKAHARA, 1990).

O reconhecimento de assinaturas em tempo real é útil principalmente em sistemas de verificação, onde um certo indivíduo terá seu acesso autorizado, por exemplo, a arquivos de um sistema operacional, através da captura e reconhecimento de sua assinatura. Em (HASTIE e KISHON et al., 1992) é proposto um sistema que captura as coordenadas pressionadas por uma caneta digitalizadora em intervalos fixos de tempo. Nesse trabalho, uma assinatura tem sua identidade reconhecida comparando-se a velocidade relativa dos movimentos feitos e a distância euclidiana entre as coordenadas das assinaturas previamente guardadas no banco de dados.

A comparação entre assinaturas deverá ser efetuada de forma a normalizar diferenças de escala e velocidade, o que normalmente é efetuado impondo a cada par de assinaturas que se pretende comparar, uma transformada conhecida como DTW *"Dinamic Time Warping"* (SANKOFF e KRUSKAL, 1999). Este tipo de transformada, também empregada em reconhecimento de voz, efetua uma compressão ou expansão de um intervalo do sinal de forma a minimizar a diferença em relação a outro. Os trabalhos (HASTIE e KISHON et al., 1992), (WOHLBERG e GREENE, 1992) e (PARIZEAU e PLAMONDON, 1990) são exemplos do uso de DTW no reconhecimento de assinaturas em tempo real.

Em certos casos, sobretudo de interesse jurídico, a informação da assinatura em tempo real não estará disponível, ou seja, o reconhecimento será executado sobre a

imagem final da assinatura. Este tipo de reconhecimento, conhecido como "offline", é geralmente mais complexo computacionalmente, pois aspectos biométricos como velocidade e pressão não estão diretamente disponíveis.

Em (SANTOS e JUSTINO et al., 2004), um sistema em "offline" é proposto onde as assinaturas são comparadas através da proporção de ocupação espacial e uma estimativa da pressão feita sobre o papel a partir da grossura das linhas de tinta. Em (PRTEUX e GENEST et al., 1997), da mesma forma, características geométricas representam a assinatura. Em (PRASAD e AMARESH, 2004), uma imagem de assinatura é comparada através da detecção do ângulo dos traços, relação entre largura e altura, área ocupada, centro de gravidade, entre outros.

2.5 Reconhecimento de voz

Os padrões de voz são influenciados pelo tamanho das cordas vocais, das cavidades nasais e de características comportamentais, sendo consideradas únicas a cada indivíduo.

A identificação de voz pode ser empregada em sistemas de verificação, onde normalmente o usuário pronuncia um texto estipulado que será comparado com outro contido em um banco de dados. Entretanto, quando o número de pessoas contidas no banco de dados é bastante grande, os sistemas de identificação de voz podem sofrer baixo desempenho computacional e não serem capazes de manter uma boa resistência a fraudes (JAIN e PANKANTI et al., 1999).

Assim como no reconhecimento de assinaturas, o caráter temporal da voz é um obstáculo para sistemas de reconhecimento ou verificação. Duas palavras ou mesmo vogais pronunciadas por um mesmo locutor terão diferentes temporizações, embora possuam características biométricas semelhantes.

Para comparar sinais de diferentes temporizações, alguns trabalhos, como por exemplo (WUTIWIWATCHAI e ACHARIYAKULPORN et al., 1999) e (SHAHIN e BOTROS, 1998), fazem uso da transformada *Dinamic Time Warping*, previamente citada na seção 2.4. Em (GERTZE, 1999) sugere-se uma implementação combinada entre redes neurais artificiais e DTW.

Dentre os trabalhos na área, diversos sistemas como (NAKAGAWA e UEDA et al., 1992), (HIGGINS e DODD et al., 2001) e (LAZLI e SELLAMI, 2003), fazem uso de modelos escondidos de markov (HMM) (RABINER, 1989). Em outros, a

preferência é pelo Modelo de misturas gaussianas (GMM) (LI e O'SHAUGHNESSY, 1997) e (SCHMIDT e GOLDEN et al., 1997) e em alguns trabalhos também são explorados as redes neurais artificiais em versões não-recorrentes (WANG e CHEN et al., 2000) e recorrentes (FRANZINI e WITBROCK et al., 1989).

Usualmente, os sistemas de reconhecimento de voz são dependentes de texto, ou seja, necessitam que o usuário fale o mesmo texto várias vezes para cadastrá-lo e repita este mesmo texto mais tarde para identificá-lo. Sistemas independentes de texto ainda representa grande desafio, embora abordado em alguns trabalhos como (ROSE e REYNOLDS, 1990) e (RUDASI, 1992), a tendência é que os sistemas independentes de texto sejam ainda menos confiáveis.

2.6 Outros reconhecimentos biométricos

Praticamente qualquer medida, fisiologia ou comportamento do corpo humano pode ser usado como método biométrico. No entanto, à exceção do reconhecimento de íris, impressões digitais, faces, voz e assinatura, a maioria dos demais métodos biométricos são empregados onde existe interesse criminal, ou seja, os métodos computacionais disponíveis têm normalmente o objetivo de simplificar o serviço de um perito.

Entre os métodos de auxilio à criminalística estão o reconhecimento de dentes, onde em (JAIN e CHEN et al., 2003) foi proposto um método semi-automático baseado em radiografias, o reconhecimento baseado em DNA, onde em (SELLERS, 1986) foi proposta uma comparação computacional. Métodos baseados em impressões digitais e voz também possuem interesse forense, como exposto nas seções 2.2 e 2.5.

De qualquer forma, existem estudos em sistemas de verificação que empregam, por exemplo, as características extraídas da geometria das mãos (JAIN e DUTA, 1999) (KUMAR e WONG et al., 2003) como método biométrico. Outros trabalhos fazem uso do contorno do corpo humano (COLLINS e GROSS et al., 2002), do estilo de andar (LEE e ELGAMMAL, 2004), da forma com que são digitadas palavras no teclado (LEGGETT e WILLIAMS, 1988) ou do formato tridimensional das orelhas de um indivíduo (CHEN e BHANU et al., 2005). Normalmente, estes métodos biométricos de verificação possuem baixo desempenho computacional, universalidade e resistência a fraudes (JAIN e PANKANTI et al., 1999).

2.7 Conclusões

Todos os sistemas biométricos apresentados neste capítulo possuem vantagens e desvantagens, que devem ser observadas previamente ao desenvolvimento de aplicações de verificação ou identificação.

Por exemplo, uma limitação do reconhecimento de íris, descrito na seção 2.1, é que os sistemas propostos necessitam ser cooperativos além de intrusivos, ou seja, exigem que a pessoa fique em uma determinada posição e apóie o queixo ou cabeça em um dispositivo para que uma câmera efetue a captação de sua íris. Sistemas não intrusivos de identificação de íris ainda possuem resultados preliminares como os mostrados em (HANNA e MANDELBAUM et al., 1996), onde um detector facial é aplicado para possibilitar a previa localização da íris.

Sem dúvida, o reconhecimento de identidade baseado em impressões digitais, detalhado na seção 2.2, é o mais adaptado a constituir prova forense, devido à sua aceitação e exclusividade. No entanto, embora a impressão digital possa ser usada de forma não cooperativa em casos onde um suspeito deixa impressões em objetos por ele manipulados, não são adaptáveis a sistemas de identificação baseados em monitoramento, pois é bastante complicado captar este tipo de padrão em fotos ou vídeo à distância. A captação é também um inconveniente em sistemas de verificação usando impressões digitais, pois há a necessidade de unidades de aquisição específicas, como as propostas em (HARTMAN, 1996).

O reconhecimento de padrões de assinatura, como descrito na seção 2.4, tem as mesmas vantagens e desvantagens do reconhecimento baseado em impressões digitais. Embora possua grande aceitação como método biométrico, principalmente forense, e seja possível desenvolver sistemas de verificação com o uso de canetas digitalizadoras, os sistemas automáticos de identificação, que necessitam funcionar em *"offline"*, ainda representam um grande desafio devido à baixa resistência a fraudes e reduzido desempenho computacional (JAIN e PANKANTI et al., 1999), além de também necessitarem ser cooperativos.

O sistemas baseados em voz, conforme descrito na seção 2.5, também devem ser cooperativos e ainda necessitam, para possuir boa confiabilidade, que o usuário fale um determinado texto para identificá-lo, o que faz com que seja difícil implementar sistemas de identificação baseados em reconhecimento de voz. O reconhecimento de faces, como descrito na seção 2.3, apresenta uma proposta vantajosa quanto a cooperatividade e facilidade de coleta. Não representa qualquer inconveniente ao usuário, pois este deverá apenas olhar para uma câmera. Em certos casos, como em câmeras de vigilância, a face é a única característica biométrica presente, sendo o padrão mais adequado para ser empregado em sistemas de identificação não cooperativos. Em contrapartida, as características de exclusividade e a permanência são relativamente mais fracas em padrões de face do que em impressões digitais ou padrões de íris, o que torna o desenvolvimento de um sistema não cooperativo ainda um desafio.

3 Detecção de Pele - Revisão Bibliográfica

Detecção de pele é uma técnica computacional que tem por objetivo identificar regiões de uma figura onde haja presença de pele humana. A detecção de pele pode ser usada em diversas áreas dentro dos métodos biométricos como a detecção de faces, sistemas que detectam a presença de pessoas, sistemas de bloqueio de conteúdo adulto (ZHENG e DAOUDI et al., 2000), entre outros. O uso da detecção de pele dentro do âmbito da detecção de faces pode ser bastante vantajoso, pois com o prévio conhecimento das regiões de uma figura que são potencialmente classificadas como pele humana, pode-se restringir a busca pela posição da face, possivelmente tornando-a mais eficiente.

A cor da pele humana é definida por fatores como a concentração de hemoglobina no sangue e a presença da proteína melanina, encontrada na maioria dos mamíferos (GEMERT e JACQUES et al., 1999). A variabilidade da cor da pele, descartadas as possíveis variações causadas por diferentes tipos de iluminação, pode ser considerada pequena, ou seja, os valores das cores dos pontos de pele têm distribuição presumivelmente compacta (JONES e REHG, 1998) e (COTTON, 1996).

3.1 Técnicas ponto a ponto

A grande maioria dos métodos encontrados para detectar regiões de pele em figuras coloridas decide se cada ponto individualmente na imagem pertence à classe de pele ou não (VEZHNEVETS e SAZONOV et al., 2003). Partindo-se do fato de que uma figura em cores possui, na codificação RGB tradicional, três valores para cada ponto em sua representação matricial, conforme mostrado na seção 5.4.1, este tipo de método deve ser capaz de decidir se cada ponto representa a pele humana utilizando apenas estes três valores.

3.1.1 Técnicas usando regras e limiares

É possível usar regras simples para definir se um ponto pertence ou não a uma região de pele humana (PEER e KOVAC et al., 2003), pois esta possui muito freqüentemente um nível de vermelho, verde e azul característicos, sendo as razões e relações entre estes valores também diferenciados mesmo entre diferentes grupos étnicos (TOMAZ e CANDEIAS et al., 2003).

Para definir estes limiares, vários autores transformam os valores de vermelho, verde e azul de cada ponto da imagem em valores supostamente melhor adaptados à detecção de pele. Essa transformação é conhecida como mudança no espaço de cores e está descrita na seção 5.4.1. Alguns trabalhos como (ALBIOL e TORRES et al., 2001) mostram que a escolha do espaço de cores não tem grande impacto no desempenho de um detector de peles.

Entre algumas destas regras simples usadas para encontrar as regiões de pele temos (PEER e KOVAC et al., 2003), mostrada no conjunto de equações 3.1, as regras sugeridas por (TOMAZ e CANDEIAS et al., 2003) são mostradas nas equações 3.2, e as regras de (ANAGNOSTOPOULOS e ANAGNOSTOPOULOS et al., 2003) são mostradas no conjunto de equações 3.3. Em todos os casos, os valores de vermelho (R), verde (G) e azul (B) estão normalizados entre 0 e 255.

$$se \ R > 95 \ e \ G > 40 \ e \ B > 20 \ e$$
$$max\{R, G, B\} - min\{R, G, B\} > 15 \ e$$
$$|R - G| > 15 \ e \ R > G \ e \ R > B \ ent \tilde{a}o \ \acute{e} \ pele.$$
(3.1)

$$Se (B > 160 e R < 180 e G < 180) ou$$

$$(G > 160 e R < 180 e B < 180) ou$$

$$(B < 100 e R < 100 e G < 100) ou$$

$$(G > 200) ou$$

$$(R + G > 400) e (G > 150 e B < 90) ou$$

$$(B/(R + G + B) > 0, 4) e (G/(R + G + B) > 0, 4) ou$$

$$(R < 102 e G > 100 e B > 110 e G < 140 ou B < 160) então é pele$$
(3.2)

 $Se \ R < 100 \ então \ não \ \acute{e} \ pele$ $Se \ R < G \ ou \ R < B \ então \ não \ \acute{e} \ pele$ $Se \ R/G > 1.3 \ e \ R/B > 1.4 \ então \ \acute{e} \ pele$

UFRJ/IM/NCE

$$Se \ R/G < 1.3 \ e \ R/B < 1.4 \ então não \acute{e} pele$$
$$Se \ R/G > 1.3 \ e \ G/B > 1.5 \ então \acute{e} pele$$
$$Se \ R/G < 1.3 \ e \ G/B < 1.5 \ então não \acute{e} pele$$
$$Se \ 77 < G < 127 \ e \ 133 < B < 173 \ então \acute{e} pele$$
(3.3)

3.1.2 Técnicas estatísticas não paramétricas

Um outro enfoque possível para detectar regiões de pele é descobrir, à priori, em um grande conjunto de dados, a probabilidade de cada cor pertencer à classe de cores de pele ou de não-pele. Ou seja, cada combinação possível entre vermelho, verde e azul terá uma probabilidade definida de pertencer a cada uma destas duas classes. Um estudo sobre este método, também conhecido como mapa de probabilidades, é mostrado em (JONES e REHG, 1998) e (BRAND e MASON, 2001). Nestes dois trabalhos, um grande conjunto de imagens teve seus pontos marcados visualmente como pele e não-pele, sendo então obtido um histograma, conforme mostrado na seção 5.4.2, de todos os pontos que foram classificados como pele e não-pele separadamente. Tendo estes dois histogramas, a probabilidade de um ponto da imagem pertencer a uma classe ou outra pode ser calculada a partir das probabilidades mostradas nas equações 3.4.

$$P(RGB = PELE) = \frac{s_{RGB}}{T_S}$$

$$P(RGB = N\tilde{A}O - PELE) = \frac{n_{RGB}}{T_N}$$
(3.4)

Onde $T_S \in T_N$ representam o número de pontos nos histogramas de pele e nãopele, respectivamente, e s e n representam o número de vezes em que a cor referida apareceu em cada um dos histogramas. Assim, para cada cor possível, ou seja, para cada combinação de vermelho, verde e azul (RGB), é calculada uma probabilidade de pertencer ao conjunto de pele e de não-pele. A partir dessas duas probabilidades, pode-se criar um critério de decisão onde um ponto é considerado pele quando a razão entre as duas probabilidades é maior do que um limiar, conforme a equação 3.5.

UFRJ/IM/NCE

$$\frac{P(RGB = PELE)}{P(RGB = N\tilde{A}O - PELE)} \ge \theta \tag{3.5}$$

Uma outra forma de estimar a probabilidade de um determinado ponto pertencer à pele humana é calcular a forma condicional P(pele|RGB), ou seja, a probabilidade de ter-se pele dado uma cor determinada. Assim, pelo teorema de bayes (JOHNSON e WICHERN, 1988) mostrado na equação 3.6, a probabilidade P(pele|RGB) pode ser calculada a partir das probabilidades estimadas nas equações 3.4, que passam a ser definidas como P(RGB|pele) e P(RGB|não-pele).

$$P(pele|RGB) = \frac{P(RGB|pele)P(pele)}{P(RGB|pele)P(pele) + P(RGB|n\tilde{a}o-pele)P(n\tilde{a}o-pele)}$$
(3.6)

Assim, para a correta estimativa de P(pele|RGB) será necessária a obtenção de P(RGB|pele) e P(RGB|não-pele). Já P(pele) e P(não-pele) podem ser obtidos através da contagem dos pontos de pele e não-pele presentes no banco de dados.

Um dos problemas que os sistemas de detecção de cor por mapa de probabilidades devem resolver é o fato de que normalmente existem muitas possíveis cores diferentes. Tendo-se 256 valores diferentes para vermelho, verde e azul, o número de diferentes cores é $256^3 = 2^{8^3} = 2^{24}$, tornando computacionalmente complexo o cálculo de P(RGB|pele) para cada uma delas. Em alguns trabalhos como (JONES e REHG, 1998), utilizou-se uma resolução de histograma com apenas 8, 16 e 32 valores possíveis para vermelho, verde e azul, percebendo-se que o sistema era mais eficiente com 32 valores e não havia vantagem em ter mais variações. Já em (BRAND e MASON, 2001), empregou-se 256 valores diferentes e as cores que por acaso não estivessem no mapa de probabilidades eram consideradas como pertencentes à classe de não-pele.

3.1.3 Técnicas estatísticas paramétricas

Uma desvantagem dos métodos não paramétricos mostrados na seção 3.1.2 está no fato de que há necessidade de se guardar uma tabela de amostras de probabilidade para cada possível cor que possa estar presente na figura em análise. No caso em

que as cores são codificadas em 8 bits por canal deve-se obter $2^{8^3} = 2^{24}$ elementos de memória para guardar as probabilidades. Uma forma de reduzir esta necessidade por memória é estimar uma função densidade de probabilidade (PDF) a ser utilizada e calcular parâmetros destas funções, de forma a ajustá-la à distribuição estimada pelos dados visualmente classificados . Vários trabalhos executam detecção de pele usando uma função gaussiana de duas ou três variáveis (SENIOR e HSU et al., 2002), (AHLBERG, 1999) e (YANG e LU et al., 1997). A base deste tipo de trabalho é usar uma gaussiana como a mostrada na equação 3.7 para estimar P(RGB|pele) a partir dos dados manualmente selecionados, a partir de imagens que contém pele.

$$p(RGB|pele) = \frac{1}{2\pi |Cov_s|^{1/2}} \cdot e^{-\frac{1}{2}(\bar{C}-\mu_s)^T Cov_s^{-1}(\bar{C}-\mu_s)^T}$$
(3.7)

Na equação 3.7, \overline{C} é o vetor de amostras de cor, que pode ter três variáveis no caso de se usar os valores de vermelho, verde e azul como mostrado na seção 5.4.1.1, ou duas variáveis no caso de não considerar a luminância (Y), ou seja, utilizar o espaço de cores YRB, conforme descrito na seção 5.4.1.2. Cov_s e μ_s são a matriz de covariância e a média, sendo então necessário escolher valores para estas matrizes de forma a estimar os dados de amostra manualmente selecionados com partes de pele. Para calcular os valores de Cov_s e μ_s podem ser usadas as equações 3.8 e 3.9, onde c_j é o vetor contendo as amostras de cor (VEZHNEVETS e SAZONOV et al., 2003).

$$\mu_s = \frac{1}{n} \sum_{j=1}^n c_j \tag{3.8}$$

$$Cov_s = \frac{1}{n-1} \sum_{j=1}^n (c_j - \mu_s) (c_j - \mu_s)^T$$
(3.9)

Alguns trabalhos como (HSUAN YANG e AHUJA, 1998) reformularam a idéia da gaussiana simples e propuseram a mistura de várias gaussianas no intuito de aumentar a eficácia do sistema de detecção de pele. Ao invés de estimar P(RGB|pele) a partir da equação 3.7, uma soma, ou mistura, de várias gaussianas é utilizada para melhor se adaptar aos dados manualmente selecionados, como mostra a equação 3.10

$$P(RGB|pele) = \sum_{i=1}^{n} \pi_i P_i(RGB|pele)$$
(3.10)

No caso da mistura gaussiana, cada P_i é uma gaussiana simples como a mostrada na equação 3.7. Dessa forma, P(RGB|pele) deve ser estimado através de um método que calcule os π_i , ou pesos, de cada uma das gaussianas, bem como Cov_s e μ_s de cada uma delas, de forma a se aproximar dos dados utilizados.

O método utilizado em (HSUAN YANG e AHUJA, 1998) e (TERRILLON e FUKAMACHI et al., 2000) é a Maximização da Esperança (ME), que encontra os valores Cov_s e μ_s para cada gaussiana, bem como o peso de cada uma delas na mistura. O número de gaussianas utilizadas na mistura varia entre os trabalhos na área, em (HSUAN YANG e AHUJA, 1998) foram usadas duas gaussianas, em (TERRILLON e FUKAMACHI et al., 2000) foram usadas oito e finalmente dezesseis em (JONES e REHG, 1998).

3.1.4 Técnicas usando redes neurais artificiais

Uma possibilidade de classificar pontos como pele e não-pele é fazer uso de uma rede neural artificial. Redes neurais são métodos computacionais que possuem a capacidade de generalização (HAYKIN, 1994), ou seja, poderão possivelmente classificar tons de pele não presentes no banco de dados de treinamento.

Um dos trabalhos mais conhecidos de detecção de pele usando redes neurais artificiais (BROWN e CRAW et al., 2001) faz uso da rede SOM (*Self Organizing Map*) (HAYKIN, 1994).

Como a rede SOM é auto-organizável, usada primariamente para clusterização (HAYKIN, 1994), a detecção de pele é feita inicialmente ajustando os pesos de uma rede SOM apenas com dados de pele. Como a rede SOM busca ajustar os pesos dos neurônios de forma a diminuir a diferença entre a entrada e os pesos sinápticos dos neurônios, após vários ciclos de ajuste, a rede SOM tenderá a ter menor erro quando for novamente apresentado um tom de pele.

Depois que uma rede SOM for organizada a partir dos dados que contém cor de pele, uma série de dados de pele e não-pele são apresentados à rede. Assim, podemse marcar os neurônios que ficam ativados quando tons de pele são apresentados e quando tons de não pele são apresentados, possibilitando a classificação (BROWN e CRAW et al., 2001).

Da mesma forma, considerando que existem duas classes possíveis no espaço de dados, as cores de pele e não-pele, existe a possibilidade de utilizar, assim como em (CHEN e ZHOU et al., 2002), uma rede neural MLP (HAYKIN, 1994) tal como mostrado na seção 5.1.1, para classificar e separar estes dados. Necessita-se para isso de um banco de imagens contendo exemplos de pele e não-pele, onde se conheça previamente a classe de cada um dos pontos presentes nas mesmas, para efetuar o treinamento da rede neural.

3.2 Uso de informações de vizinhança em métodos de detecção de pele

O uso de informações de vizinhança pode tornar mais eficientes os métodos ponto a ponto descritos na seção 3.1. A decisão de que um ponto representa pele não é tomada apenas sobre os valores RGB do ponto em teste, mas também afetada pelos pontos vizinhos.

Sabe-se que regiões de pele formam agrupamentos na maioria das figuras, ou seja, pontos de pele estarão freqüentemente cercados de vizinhos também classificados como pontos de pele. Métodos puramente ponto a ponto não aproveitam estas informações de conectividade, podendo decidir por exemplo, que um ponto isolado pertence à classe de pele enquanto todos os seus vizinhos são considerados não-pele, o que constitui certamente um erro.

Em (DEL SOLAR e VERSCHAE, 2004), usa-se a informação de vizinhança para melhorar a eficiência do método de mistura gaussiana descrito na seção 3.1.2. Neste trabalho, depois do processo de mistura gaussiana, os pontos serão considerados como pele se todos os seus oito vizinhos também obtiveram alta probabilidade de pertencer a classe de pele. Além disso, a distância gaussiana entre os valores RGB do ponto em teste e seus vizinhos deve ser menor que um limiar indicado pelo usuário.

Outro exemplo de uso da informação de vizinhança é encontrado em (TOMAZ e CANDEIAS et al., 2003) e (FERIS e DE CAMPOS et al., 2000). Embora sejam

UFRJ/IM/NCE

utilizados métodos ponto a ponto, a informação de conectividade é aproveitada pois um filtro de mediana é aplicado sobre a decisão de pele e não-pele. Este filtro manterá como pele apenas os pontos cuja maioria dos vizinhos seja também classificado como pele.

3.3 Conclusões

As classes de pele e não-pele não serão totalmente separáveis apenas por seus valores RGB, por melhor que sejam os sistemas de avaliação, pois não há qualquer impedimento de existir objetos da mesma cor característica de pele, ou até mesmo que o fundo tenha cor de pele.

A detecção de pele ponto a ponto também será influenciada pela iluminação ambiente, que poderá fazer com que uma área de pele fique muito escura ou demasiadamente clara, aproximando-se do branco ou preto 100 % que obviamente não representam cor de pele. Para evitar este problema, algum tipo de correção de iluminação deverá ser aplicado.

Um sistema de detecção de pele também deve possuir algum pós-processamento, para avaliar a vizinhança de cada ponto detectado como pele. Devido à continuidade das áreas de pele na figura, os pontos detectados tendem a agrupar-se em blocos, o que poderá ser aproveitado para eliminar falso-positivos.

Mesmo constituindo um grande desafio, a detecção de pele apresenta a grande motivação de permitir uma diminuição no espaço de busca pela face. Nesta dissertação um sistema de detecção de pele baseado em redes neurais MLP é proposto, como descrito na seção refsec:contribuicoes, onde também são avaliados métodos de pré e pós-processamento.

4 Detecção de Faces - Revisão Bibliográfica

Detectar faces é um método, ou conjunto de métodos, capazes de encontrar a posição de uma face humana em uma figura maior, contendo qualquer tipo de fundo e em qualquer posição. A detecção de faces não deve ser confundida com o reconhecimento facial, cujo trabalho é descobrir a identidade de uma face, embora freqüentemente ambos os sistemas sejam encontrados juntos.

Muitos trabalhos buscaram, ao longo dos anos, tornar a detecção de face imune às variações de rotação, tamanho, iluminação e oclusão parcial, tornando a detecção de faces uma tarefa tão complexa quanto o próprio reconhecimento de identidade.

4.1 Motivação e aplicações

Qualquer sistema de reconhecimento facial precisa, inicialmente, separar a região da face do fundo e de outros objetos presentes na imagem. Freqüentemente, em ambientes não controlados, esta é uma tarefa complexa, necessitando de um estudo específico para cada caso.

Uma aplicação mais importante para a detecção facial é, a primeira vista, sistemas de reconhecimento de faces. Separar com sucesso a região da face da figura maior é o primeiro passo para um reconhecimento correto.

No entanto, existem outras aplicações possíveis a serem exploradas como por exemplo: detectores de presença ou gravação disparados por faces em sistemas de segurança, compressão de imagens e vídeo com taxa variável (com melhor taxa na região das faces) além da catalogação de imagens na Internet.

Existem muitos trabalhos sobre detecção facial, usando diversas técnicas desde a simples detecção de cores de pele até métodos neurais de procura por padrões na imagem. Há dois tipos básicos de detecção de faces, os métodos de detecção facial por características, onde existe uma preocupação em encontrar formas, cores ou detalhes próximos a uma face humana, estudadas a priori, e os métodos baseados na imagem propriamente dita, que faz uso de técnicas de reconhecimento de padrões como as redes neurais ou métodos probabilísticos.

4.2 Métodos baseados em características faciais

Métodos de detecção baseados em características da imagem foram os primeiros métodos de detecção facial. Estes tipos de métodos necessitam de informação sobre a face, normalmente conseguida por observação e estudo das características típicas da face humana como cor, forma, proporção, posição ou movimento. Estão incluídos nestes métodos a detecção por bordas, por características colorimétricas, por movimento ou modelos de face.

4.2.1 Detecção por bordas

O mais simples tipo de detecção de objetos é a detecção por bordas, usada pela primeira vez na detecção facial em (SAKAI e NAGAO et al., 1972), um dos mais antigos trabalhos da área, que utiliza operadores de extração de bordas como o *Sobel* ou *Laplacian* (JAIN, 1989), e busca comparar o resultado com padrões previamente conhecidos de face. Um dos padrões usados para validação é a razão de ouro (FARKAS e MUNRO, 1987) mostrada na equação 4.1, usada também por artistas para manter uma proporção natural da face em desenhos e ilustrações.

$$\frac{Altura\ da\ face}{largura\ da\ face} \cong \frac{1+\sqrt{5}}{2} \tag{4.1}$$

Entre os trabalhos que usam detecção de face por bordas estão (GOVINDARAJU, 1996) e (HERPERS e KATTNER et al., 1995), ambos comparando as bordas retiradas da imagem original com padrões antropométricos, ou seja, padrões comuns da face humana estudados na medicina, como os mostrados em (FARKAS e MUNRO, 1987).

Uma outra possibilidade é uma procura por densidade de bordas, onde são verificadas as regiões que possuem mais pontos de bordas. A região dos olhos normalmente é cercada por áreas que possuem muitas variações dos valores de luminância (DE SILVA e AIZAWA et al., 1995). Áreas mais escuras como sobrancelhas, cílios e pupilas, estão relativamente próximas de áreas mais claras como a esclerótica, a íris e as pálpebras. Estas variações de luminância podem ser detectadas como bordas, constituindo um modelo simples para encontrar a região da face. Este tipo de detecção pode ser melhorado com o uso de modelos de face para validação, como mostrado na seção 4.2.4.

UFRJ/IM/NCE

As figura 4.1 mostra um exemplo da extração de bordas na detecção de faces. Caso um sistema automático esteja a procura da região dos olhos ou da face, a procura por pontos que possuem bordas pode diminuir o espaço de busca que do contrário seria a figura toda, o que é possível pois a região dos olhos estará muito possivelmente cercada por regiões de bordas.



Figura 4.1: Exemplo de extração de bordas, a busca pelos olhos torna-se mais fácil pois o espaço de busca foi diminuído.

4.2.2 Detecção por características colorimétricas

Usar a informação de cores para detectar face também é bastante comum como mostrado em (KJELDSEN e KENDER, 1996), (YANG e WAIBEL, 1996) e (MCKENNA e GONG et al., 1996). A idéia mais simples da detecção de faces por cor é usar um dos métodos descritos na seção 3 para encontrar as prováveis regiões de pele humana presentes na figura original, e a partir destas regiões encontrar particularidades da face como a forma ou a disposição das partes de pele em relação a posição dos olhos, boca ou sobrancelhas.

Muitos sistemas, como (KRUPPAAND e BAUER et al., 2002), executam a detecção facial por características colorimétricas de forma rápida e eficiente. Logicamente, considerando que não há restrição de fundo ou iluminação nas figuras utilizadas, a detecção de pele poderá encontrar alguns falsos positivos, que no caso de (KRUPPAAND e BAUER et al., 2002) são retirados por uma análise de formas, feita para assegurar que apenas as regiões com forma elíptica sejam aproveitadas. Um exemplo de detecção de face deste trabalho é mostrado na figura 4.2.

UFRJ/IM/NCE



Original

Detecção de pele

Detecção de pele filtrada pela análise de formas

Figura 4.2: Exemplo de detecção de faces usando características colorimétricas, além de uma análise de formas faciais elípticas (KRUPPAAND e BAUER et al., 2002). (Em cores)

Vale ressaltar que as informações de cor de uma imagem não são normalmente utilizadas como único método de detecção facial, mas sim como apoio a outros métodos, como forma de diminuir a região de procura pela face, partindo do princípio lógico que a mesma muito possivelmente terá cor de pele detectável pelos sistemas da seção 3. Esta mesma idéia, ou seja, diminuir o espaço de busca, também é aplicada na detecção por bordas, da seção 4.2.1.

4.2.3 Detecção usando movimento

Detectar a face pelo movimento pode ser possível se arquivos de vídeo estão disponíveis. Em imagens frontais, movimentos da boca podem ser usados para encontrar regiões candidatas a face (GRAF e COSATTO et al., 1996).

Também em imagens frontais, a constante repetição do piscar dos olhos pode ser usada para encontrar regiões candidatas a face (MORRIS, 2002).

Assim como a detecção por características colorimétricas, mostrada na seção 4.2.3, a detecção usando movimento pode também ser usada para encontrar regiões candidatas a face, e evitar assim que o sistema precise procurar faces em todas as regiões dos novos frames capturados por uma câmera. Técnicas de detecção de movimento podem ser encontrados em (JAIN, 1989).

4.2.4 Detecção por modelos faciais

Características antropométricas da face são estudadas em diversos campos da medicina. Algumas destas características, mostradas na figura 4.3, podem ser usadas para simplificar a detecção facial, as distâncias entre o ponto central n ou OR e os demais, bem como suas devidas variações, estão catalogadas em (FARKAS e MUNRO, 1987).



Figura 4.3: Diversas medidas antropométricas da face (FARKAS e MUNRO, 1987)

Um exemplo de detecção por modelo pode ser encontrado em trabalhos como (BEEK e REINDERS et al., 1992), (CROWLEY e BERARD, 1997) e (GRAF e CHEN et al., 1994). Estes utilizam a posição dos olhos e boca como referência de medida para encontrar a posição da face, principalmente em figuras que possuem apenas uma pessoa. Inicialmente é feita uma detecção de bordas ou uma filtragem passa-banda específica, seguido de uma procura por objetos cuja razão espacial x/y mais se aproxime dos olhos e da boca, que possuem uma forma específica, como ilustrado na figura 4.4.

Logicamente, a busca por objetos que tenham a razão espacial mostrada na figura 4.4 pode originar em vários pontos que não necessariamente representam olhos humanos. A partir então destes diversos pontos, a posição correta dos olhos e da boca é feita através de medidas antropométricas, sabendo-se que a posição da boca estará abaixo dos olhos na horizontal, e entre os mesmos na vertical, além do fato de que os olhos deverão estar alinhados, já que este trabalho trata de fotos frontais.

UFRJ/IM/NCE



Figura 4.4: As razões $\frac{X_{boca}}{Y_{boca}}$ e $\frac{X_{olho}}{Y_{olho}}$ são diferentes, o que permite separar os olhos da boca

Alguns trabalhos utilizam um modelo geral da face, baseados em medidas antropométricas, e buscam adaptá-lo a figura em que se deseja detectar a face. Em (DE SILVA e AIZAWA et al., 1995), por exemplo, os olhos são inicialmente detectados usando detecção por densidade de bordas como mostrado na seção 4.2.1, e a seguir um padrão de face é ajustado comparativamente usando uma função de custo proporcional ao número de pontos de borda comuns entre o modelo e a figura original. O padrão de face deste trabalho (DE SILVA e AIZAWA et al., 1995) foi conseguido após a medição sistemática de várias figuras de face.

4.3 Métodos baseados na imagem

Métodos de detecção baseados na imagem são mais recentes que os baseados em características, e buscam encontrar padrões de face com técnicas probabilísticas, redes neurais e outros. Tendem a apresentar melhores resultados mas são computacionalmente mais complexos. A maioria dos trabalhos atuais se encaixa neste tipo de detecção.

4.3.1 Métodos de detecção facial usando redes neurais

Redes neurais artificiais (RNA), descritas na seção 5.1, constituem um conhecido método de reconhecimento de padrões, sendo sua aplicação em detecção de faces

avaliada principalmente em (ROWLEY e BALUJA et al., 1996) e (ROWLEY e KANADE, 1999), onde são utilizadas redes neurais MLP, abordadas na seção 5.1.1.

O método de detecção usando redes neurais tem por base um banco de dados de pequenas imagens de face recortadas e centralizadas cuidadosamente para que a posição dos olhos, nariz e boca variem muito pouco entre as imagens.

Dessa forma, uma rede neural artificial com dois neurônios de saída recebe as imagens, e é treinada para permitir ativação de um dos neurônios quando figuras de face são apresentadas, ou seja, figuras que tem a disposição típica de face, com olhos, nariz e boca nas posições adequadas. Um banco de dados composto de ruído e figuras geométricas aleatórias que não representam faces é apresentado à rede de forma a ativar o outro neurônio, indicando que não se trata de uma face.

O método neural de detecção de faces, originalmente proposto por (ROWLEY e KANADE, 1999), foi selecionado para compor o sistema detector de faces usado nesta dissertação. Por este motivo, mais detalhes sobre o mesmo, bem como as modificações propostas, são apresentados na seção 6.3.

4.3.2 Métodos de detecção facial usando grafos

A detecção de face usando grafos foi originalmente proposta por (WISKOTT e FELLOUS et al., 1996) e (WISKOTT e FELLOUS et al., 1997), e consiste em criar um grafo elástico cujos nós estão em pontos que representam particularidades da face como o contorno da boca, as bordas dos olhos, o centro do queixo, etc.

Em (WISKOTT e FELLOUS et al., 1997), propõe-se que os vértices do grafo sejam encontrados em uma nova face através do cálculo de uma similaridade destes com um banco de dados previamente montado com outras figuras de faces.

Para montar este banco de dados, inicialmente aplica-se a convolução da imagem original, como mostra a equação 4.2 com uma função de gabor mostrada na equação 4.3.

$$J_{j}(\vec{x}) = \int I(\vec{X}) \Psi_{j}(\vec{x} - \vec{X}) d^{2} \vec{X}$$
(4.2)

$$\Psi_j(\vec{x}) = k_j^2 \exp\left(\frac{-k_j^2 x^2}{2\sigma^2}\right) \left[\exp(i\vec{k}_j \vec{x}) - \exp(-\frac{\sigma^2}{2})\right]$$
(4.3)

Sendo $\vec{x} = \{x, y\}$ e *I* respectivamente a localização e luminância de um ponto da imagem. O vetor \vec{k}_j em 4.3 é dado pela equação 4.4, e tem os valores $\nu \in \mu$ representando respectivamente a freqüência e orientação da função de gabor.

$$\vec{k}_j = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_\nu \cos(\varphi_\mu) \\ k_\nu sen(\varphi_\mu) \end{pmatrix}, k_\nu = 2^{-\frac{\nu+2}{2}}\pi, \varphi_\mu = \frac{\pi}{8}$$
(4.4)

Em (WISKOTT e FELLOUS et al., 1997) propõe-se que a equação 4.2 seja calculada para 5 diferentes valores de ν e 8 valores de μ na equação 4.4, o que resultará em quarenta diferentes valores de J_j para cada ponto da imagem. Estes quarenta valores formam o chamando Jet, que contém características das particularidades da face.

A extração dos Jets é considerada uma transformada de wavelet (BURRUS e GOPINATH et al., 1998), onde a equação 4.3 é a função mãe, que é comparada com a imagem em diferentes escalas e rotações.

Finalmente, um grande número de imagens de faces tem as mesmas particularidades marcadas manualmente. Em (WISKOTT e FELLOUS et al., 1997) dezesseis particularidades são marcadas, como mostra a figura 4.5. Para cada destas particularidades em cada imagem no banco de dados são calculados os Jets.



Figura 4.5: Dezesseis particularidades marcadas para cálculo dos Jets

Assim, de acordo com a equação 4.5, cada jet será composto por quarenta coeficientes complexos com amplitudes a_i e fases ϕ_i

$$J_j = a_j \exp(i\phi_j) \tag{4.5}$$

Os Jets podem ser comparados de acordo com a sua similaridade, a equação 4.6 mostra o cálculo de similaridade de dois Jets. A fase ϕ_j pode ser ignorada no cálculo de similaridade, como mostra a equação 4.6, ou pode ser feita uma compensação. Mais detalhes em (WISKOTT e FELLOUS et al., 1997).

$$S_a(J, J') = \frac{\sum_{j} a_j a'_j}{\sqrt{\sum_{j} a_j^2 \sum_{j} a'_j^2}}$$
(4.6)

Quando uma nova imagem é apresentada, um grafo com tamanho padronizado é colocado em posição arbitrária e tem seus jets calculados. Estes têm sua similaridade comparada com todas as imagens no banco de dados. Aquela que obtiver maior similaridade S_a será usada como padrão de comparação.

Inicialmente, o grafo padronizado é ajustado como um todo, ou seja, todos os seus nós são movidos ao mesmo tempo, buscando o maior somatório das similaridades de todos os jets das particularidades. Este ajuste pode ser feito usando um algoritmo de busca como o gradiente descendente.

Em um segundo momento, os grafos podem ter seus nós movidos independentemente, possibilitando um ajuste mais fino, buscando aumentar a similaridade dos jets. Ao final do processo, a face estará detectada em conjunto com a localização de diversas particularidades, que podem fornecer informações para um algoritmo de reconhecimento de identidade.

Entretanto o algoritmo descrito em (WISKOTT e FELLOUS et al., 1997) pressupõe que o fundo não contém outros objetos que poderiam causar perturbações no posicionamento dos grafos apresentados. Além disso, não se adapta para o caso onde possam existir N faces na figura, sendo melhor indicado para captura de imagens com fundo controlado, para aplicação direta de sistemas de reconhecimento de identidade.

4.4 conclusões

A maioria dos sistemas de detecção de faces apresentados nesta seção foi desenvolvido para faces frontais, sendo normalmente pequenas variações permitidas sem grandes prejuízos ao desempenho dos sistemas.

Devido a grande variabilidade do padrão de face, principalmente sem a que face esteja necessariamente frontal, o desenvolvimento de um sistema detector de faces em qualquer situação seria um grande desafio computacional.

Mesmo que as faces sejam frontais, o que torna o padrão menos variável, ainda deve-se discutir possíveis rotações, o que também constitui uma grande dificuldade. Não há também qualquer impedimento de que imagens possuam objetos semelhantes ao padrão de face.

Como um sistema de detecção de faces é proposto nesta dissertação, como mostra a seção 6, mais detalhes serão apresentados sobre suas características e funcionalidade, incluindo também resultados e uma avaliação de desempenho, como será mostrado na seção 7.

5 Fundamentação teórica

Neste capítulo será apresentada uma breve fundamentação teórica sobre as técnicas e modelos utilizados nesta dissertação.

5.1 Redes neurais artificiais

Uma rede neural artificial (RNA) é um método computacional capaz de simular a capacidade do cérebro humano de reconhecer padrões, aprender e generalizar. A principal característica de uma RNA é a sua habilidade de adquirir conhecimento, adaptando-se a situações que não fizeram parte do conjunto de treinamento (HAYKIN, 1994).

Uma RNA consiste em uma ou várias unidades processadoras interligadas, conhecidas como neurônios artificiais, dispostas em uma ou mais camadas. Cada camada pode conter diversos elementos processadores. A conexão entre as unidades processadoras é feita através dos pesos sinápticos que, em outras palavras, definem a influência entre as unidades interligadas. Esses pesos, que são parâmetros internos da rede, têm seu valor atualizado durante um processo iterativo chamado treinamento.

O treinamento de uma RNA pode ser feito de forma supervisionada, onde o ajuste iterativo dos pesos é feito de forma a obter-se da rede neural um sistema de classificação ou previsão, ou de forma não supervisionada, onde o ajuste de pesos possibilita uma clusterização dos dados de entrada.

5.1.1 O modelo MLP

Uma das redes neurais mais encontradas em trabalhos da área de inteligência computacional é o modelo *multi-layer perceptron* ou MLP. Este modelo possui treinamento supervisionado e foi empregado nesta dissertação na detecção de indícios da face, descrito na seção 6.1 e na detecção de faces propriamente dita, descrita na seção 6.3.

A rede neural MLP é formada por camadas de neurônios, de acordo com o modelo mostrado na figura 5.1. As entradas, representadas por $P_{1..R}$ são os estímulos a serem analisados pela rede e para cada entrada há um peso sináptico associado, representado por $W_{1..R}$.



Figura 5.1: Modelo de neurônio artificial

O neurônio artificial então efetuará o somatório de todas as entradas ponderadas por seus devidos pesos, sendo somado um valor de bias (HAYKIN, 1994), representado por b_j na figura 5.1. Sobre este somatório é aplicada a função de propagação (THOMÉ, 2005), que tem por objetivo definir a forma com que um neurônio será ativado pelas entradas, como é mostrado no conjunto de equações 5.1.

$$net_j = \sum_{i=1}^R w_r p_r + b_j$$

$$y_j = f(net_j)$$
(5.1)

Quando neurônios são dispostos em apenas uma camada, como mostrado na figura 5.2, forma-se uma estrutura conhecida como perceptron (MCCULLOCH e PITTS, 1943), que possui a capacidade de classificar dados que sejam linearmente separáveis. A função de propagação utilizada no perceptron (ROSENBLATT, 1958) é um degrau, ou seja, age como um limiar que faz com que o neurônio seja ativado $(y_j = 1)$ quando o valor do somatório net_j for maior que um determinado valor e desativado $(y_j = 0)$ caso contrário.

O modelo MLP é formado por várias camadas de perceptrons, o que significa que as entradas $P_{1..R}$ podem representar estímulos externos ou internos (saídas dos neurônios da camada anterior), como mostra a figura 5.3. As camadas que ficam



Figura 5.2: Perceptron, apenas uma camada de neurônios

entre a entrada e a camada de saída são conhecidas como camadas escondidas, o que faz a rede neural MLP da figura 5.3 possuir apenas uma camada escondida.



Figura 5.3: Modelo MLP

Cada unidade processadora, ou neurônio, de um MLP executa uma operação simples sobre a informação recebida da entrada externa ou da saída de outro neurônio. O resultado é então transmitido para outros neurônios que, por sua vez, repetem a operação. O processo de atualização dinâmica dos pesos, conhecido como treinamento, é feito através do algoritmo conhecido como retro-propagação de erro (RUMELHART e DURBIN et al., 1993).

O algoritmo de retro-propagação propõe que o treinamento da rede neural seja feito em duas etapas a cada iteração. Na primeira, o sinal de entrada é entregue aos neurônios da primeira camada, que propagam o estímulo recebido até os neurônios de saída. A seguir, o erro $e_j(n)$ é calculado como mostra a equação 5.2, através da diferença entre a saída obtida $y_j(n)$ e a desejada $d_j(n)$, onde n é o número da iteração.

$$e_j(n) = d_j(n) - y_j(n)$$
 (5.2)

O ajuste de cada peso sináptico da rede MLP deve ser calculado de acordo com a derivada parcial do erro total da rede, que é o somatório e_{total} dos quadrados dos erros de todos os neurônios de saída, em relação aos pesos (ZURADA, 1999), como mostrado na equação 5.3. O peso ajustado em 5.3 é w_{ji} , que corresponde à uma ligação sináptica entre a saída de um neurônio na camada *i* e entrada de um neurônio na camada *j*. O sinal negativo indica que os ajustes de peso estarão no sentido contrário ao crescimento do erro, enquanto η é um parâmetro de treinamento chamado taxa de aprendizado.

$$\Delta w_{ji}(n) = -\eta \frac{\delta e_{total}(n)}{\delta w_{ji}(n)}$$
(5.3)

A derivada $\frac{\delta e_{total}(n)}{\delta w_{ji}(n)}$ pode ser expressa pela equação 5.4 através da regra da cadeia, onde p_j é a entrada aplicada ao peso w_{ji} e f' é a derivada da função de propagação.

$$\frac{\delta e_{total}(n)}{\delta w_{ji}(n)} = -e_j \cdot f'(net_j) \cdot p_j \tag{5.4}$$

O que faz com que o ajuste de peso seja dado por 5.5, onde $\delta_j(n)$ é conhecido como gradiente local, e indica as modificações necessárias nos pesos.

$$\Delta w_{ji} = \eta \delta_j(n) a_i$$

UFRJ/IM/NCE

$$\delta_j(n) = e_j f'(net_j) \tag{5.5}$$

A regra mostrada em 5.5 pode ajustar os pesos dos neurônios da camada de saída, mas não é capaz de ajustar pesos de neurônios nas camadas escondidas, pois neste caso o valor do erro e_j não poderá ser calculado diretamente pela equação 5.2, já que não existe valor desejado para a saída dos neurônios das camadas ocultas. Neste caso, o gradiente local deve ser calculado de acordo com a equação 5.6.

$$\delta_j(n) = f'(net_j) \sum_k \delta_k(n) w_{kj}(n)$$
(5.6)

Onde agora j refere-se a um neurônio em camada anterior a k, ou seja, o neurônio está escondido. Para calcular o valor do gradiente $\delta_j(n)$ necessita-se calcular o valor dos gradientes locais de todos os neurônios na camada k e efetuar o cálculo do somatório da equação 5.6.

Em resumo, o algoritmo de retro-propagação deve calcular o gradiente local através da equação 5.5 para os neurônios de saída ou pela equação 5.6 para os neurônios das camadas escondidas, e a seguir obter o ajuste Δw_{ji} dos pesos através da equação 5.3. Este processo deve ser repetido *n* vezes até que o erro seja tão baixo quanto necessário, onde é terminado o treinamento.

Conforme se verifica na equação 5.3, o ajuste de peso deve ser efetuado em conjunto com o cálculo do valor da derivada $f'(net_j)$. Isso impõe que a função de propagação seja derivável, não possibilitando o uso de um limiar como no perceptron. As funções de ativação mais utilizadas em redes MLP são a log-sigmóide ou a tan-sigmóide (DEMUTH e BEALE, 2001), mostradas nas equações 5.7 e 5.8 respectivamente. O valor de *a* nas curvas define a inclinação, como mostram os gráficos da figura 5.4.

$$f(net_j) = \frac{1}{1 + \exp(-a \cdot net_j)}; \ a > 0$$
(5.7)

$$f(net_j) = tanh(a \cdot net_j); \ a > 0 \tag{5.8}$$

A taxa de aprendizado η é um importante parâmetro no treinamento da rede neural. Se η for pequeno demais, o treinamento poderá ser lento e não conseguir transpor pontos de mínimo local. Em contrapartida, um valor de η elevado demais


Figura 5.4: Gráficos das funções de ativação mais usadas em redes MLP, log-sigmóide à esquerda e tan-sigmóide à direita. (Em cores)

pode causar oscilações no treinamento, não permitindo uma convergência adequada. Uma das variações do algoritmo básico de retro-propagação é a aprendizagem adaptativa, onde o valor de η é aumentado enquanto o erro estiver decrescendo e diminuída caso o contrário, o que possibilita uma rápida convergência inicial evitando oscilações ao final do treinamento.

Devido a tornar-se computacionalmente lento, principalmente com o aumento da dimensão e complexidade dos dados de treinamento, o algoritmo de retro-propagação foi modificado dando origem a vários métodos alternativos como o gradiente conjugado (HAYKIN, 1994), a propagação resiliente (DEMUTH e BEALE, 2001) e o quickprop (FAHLMAN, 1988), entre outros, que se tornaram mais adequados para problemas onde a dimensão dos dados de entrada é elevada.

Nesta dissertação, os treinamentos de redes neurais são executados usando propagação resiliente, que se mostrou mais adequado aos dados e problemas apresentados. No entanto, verificou-se que os treinamentos executados com gradiente conjugado ou retro-propagação tradicional não resultam em qualquer ganho significativo no desempenho da rede neural.

5.1.2 Validação e parada antecipada

O treinamento de uma rede neural normalmente termina quando atingido o valor objetivo de erro ou o máximo de épocas. Entretanto, se este objetivo for pequeno demais, pode ocorrer um efeito chamado *"overfitting"* (HAYKIN, 1994) (DEMUTH e BEALE, 2001).

Quando um treinamento de rede neural está em *"overfitting"*, embora o valor do SSE do banco de dados de treinamento continue a diminuir, a generalização diminui, ou seja, o SSE em um banco de dados não visto pela rede neural tenderá a aumentar.

Para evitar que isto ocorra, vários métodos de treinamento sugerem uma parada antecipada, avaliando-se o SSE de um banco de dados à parte chamado validação. A validação não é utilizada para ajustar os pesos da rede neural durante o treinamento, mas a cada época avalia-se o SSE da validação, para verificar se ocorreu *"overfitting"*.

Uma das formas de detectar "overfitting" pela validação é verificar o número de épocas em que o SSE da validação não diminui. Caso este número de épocas seja maior que um valor de limiar, deve-se parar o treinamento e voltar ao ponto onde o SSE da validação foi mínimo, ou seja, o ponto onde se acredita que a rede obterá a melhor generalização.

A figura 5.5 mostra um exemplo deste processo. Em azul está o SSE do vetor de treinamento, que decresce com o número de épocas. Em verde está o SSE do vetor de validação, que também decresce com o número de épocas até o ponto representado pelo círculo verde. A partir deste ponto, pode-se dizer que ocorre *"overfitting"*, pois embora o SSE do banco de dados de treinamento continue a diminuir, observa-se uma deterioração na capacidade de generalização da rede, representada pelo SSE da validação.

5.1.3 Uso de Componentes Principais (PCA) em Redes Neurais

Uma análise de um banco de dados real possivelmente revelará uma alta correlação entre as variáveis existentes. Isto dificulta o treinamento de uma rede neural e sua capacidade de generalização e classificação, pois se n variáveis são correlacionadas a informação carregada por uma delas tem grande probabilidade de ser semelhante às outras, isto faz com que os algoritmos de treinamento necessitem envolver-se com a complexidade de um grande conjunto de variáveis sem que estas sejam realmente relevantes.

Esta observação é especialmente verdadeira no caso de imagens, em que cada variável representa, por exemplo, um nível de cinza na figura. Sabe-se que vários sistemas de compressão utilizam-se do fato de que um pixel na imagem é possivel-



Figura 5.5: Exemplo de uso do banco de dados de validação para parada antecipada no ponto de melhor generalização. (Em cores)

mente semelhante aos demais, ou seja, um pixel é altamente correlacionado com seus vizinhos.

A técnica de PCA "Principal Components Analysis", ou análise de componentes principais, tem por objetivo criar uma transformação linear P_{trans} que projete a matriz de dados p de forma a descorrelacionar suas variáveis. Esta técnica também é conhecida como transformada de Karhunen-Loève ou transformada de Hotelling.

Para efetuar esta transformação linear, o algoritmo de componentes principais calcula inicialmente a matriz de covariância ρ_p de p. A seguir, é efetuado o cálculo dos autovalores λ_N e autovetores q_N da matriz de covariância (HAYKIN, 1994).

Os N autovetores q_N da matriz de covariância, onde N é o número de dimensões de p, representarão os eixos escolhidos para tornar as variáveis descorrelacionadas (JOHNSON e WICHERN, 1988). A figura 5.6 mostra um exemplo do cálculo dos autovetores a partir de uma distribuição criada artificialmente. Nota-se que há claramente uma correlação entre as variáveis x e y da figura 5.6, e que existem dois clusters, que neste caso indicam classes que se deseja separar.



Figura 5.6: Distribuição criada artificialmente para possuir correlação. Em vermelho são plotados os autovetores q_N da matriz de covariância (PCA), onde $\lambda_1 e \lambda_2$ indicam os autovalores associados. (Em cores)

Uma transformação linear deverá ser aplicada para permitir que os autovetores passem a representar os eixos da distribuição. Nesta transformação, novas variáveis serão compostas por parcelas ponderadas das variáveis de p (HAYKIN, 1994). O resultado final da distribuição será semelhante a uma rotação de forma a trazer o autovetor de maior autovalor (q_2 no caso da figura 5.6) para o eixo dos X.

Os autovalores λ_N associados aos autovetores indicam a variância de cada nova variável no espaço projetado (HAYKIN, 1994). Assim, caso os autovalores sejam apresentados em ordem decrescente, normalizados de forma que o seu somatório represente a unidade, pode-se estimar um limiar de perdas aceitáveis para diminuição da dimensionalidade.

Considerando que a soma de todos os autovalores até M, onde M < N, é igual a 0,9, entende-se que 90% da variância será mantida se todos as variáveis de M + 1 a N do espaço projetado forem eliminadas. Esta redução de dimensionalidade poderá resultar em treinamentos mais rápidos e melhor generalização da RNA (HAYKIN, 1994)

A figura mostra a transformação linear já aplicada à distribuição da figura 5.6, as classes $C_1 \in C_2$ podem ser separadas apenas com a variável X da distribuição, indicando que a variável Y não carrega informação relevante, o que também poderia ser verificado através do autovalor λ_1 , que é bem menor que λ_2 .



Figura 5.7: Distribuição da figura 5.6 rotacionada de forma que os eixos sejam os autovetores da matriz de covariância (PCA). (Em cores)

A técnica de PCA foi empregada nesta dissertação para diminuição de dimensionalidade nos treinamentos das redes neurais detectoras de faces, como descrito na seção 6.3.2.

5.2 Método de clusterização K-means

Métodos de clusterização devem ser capazes de dividir N pontos em K classes ou clusters, minimizando algum critério de pertinência.

O algoritmo K-means é um dos mais simples métodos de clusterização, onde os N pontos serão separados em K clusters cujo centro será C_K (GNANADESIKAN, 1977). O critério a ser minimizado será a diferença entre cada ponto e o centro do cluster, como definido pela função f(K, N) da equação 5.9.

$$f(K,N) = \sum_{j=1}^{K} \sum_{i=1}^{N} |X_i - C_j|^2$$
(5.9)

A cada passo o algoritmo K-means calculará a distância de cada ponto N até cada centro C_K , colocando-o no cluster mais próximo, ou seja, em que a distância

 $|X_i - C_j|^2$ for menor. Durante a implementação, usualmente é definida uma matriz binária $M_{K \times N}$ conhecida como matriz de pertinência, que contém a informação de quais pontos pertencem a quais clusters (SPATH, 1985).

O algoritmo finaliza a clusterização quando entre dois passos seguidos não houver mudança na matriz de pertinência. Desta forma, o algoritmo K-means pode ser definido pelos seguintes passos.

- 1. Inicializar aleatoriamente a matriz de pertinência $M_{K \times N}$, ou seja, colocar de forma aleatória cada ponto N em um dos K clusters.
- 2. Calcular os centros C_K através da média dos valores das coordenadas dos pontos designados no cluster K pela matriz de pertinência. Esta operação pode ser vista como um cálculo de centro de massa dos pontos de cada cluster.
- 3. Redefinir a matriz de pertinência $M_{K \times N}$ colocando cada ponto N no cluster K cujo centro C_K estiver mais próximo.
- 4. Verificar se houve mudança na matriz de pertinência, caso não exista, o algoritmo termina, senão, retorna para o passo de número 2.

A figura 5.8 mostra a evolução de um algoritmo K-means aplicado em uma distribuição criada artificialmente. As cores indicam a pertinência e os círculos o centro dos clusters. Nota-se que no estado inicial a pertinência é definida aleatoriamente, mas com apenas 7 iterações o algoritmo converge para o correto agrupamento da distribuição.

Vale ressaltar que como a matriz de pertinência é inicializada aleatoriamente, cada execução do algoritmo pode tender a uma diferente clusterização dos pontos (SPATH, 1985).

Nesta dissertação, escolheu-se o algoritmo K-means devido a sua simplicidade e rapidez de convergência, sendo empregado com o objetivo de tornar mais eficaz a redução de pontos de borda-pele em distribuições mais complexas, como será descrito na seção 6.2.1.



Figura 5.8: Exemplo da evolução do algoritmo K-means com K = 6 clusters, aplicado em uma distribuição criada artificialmente. (Em cores)

5.3 Algoritmos genéticos

Um algoritmo genético é basicamente um conjunto de métodos de maximização de funções motivados pela forma com que a evolução adapta os indivíduos às necessidades do meio-ambiente.

Em um algoritmo de maximização, deseja-se encontrar os valores de X_i (i = 1, 2, 3, ...) em que a função $f(X_i)$ seja máxima. Diversos métodos se propõem a encontrar estes máximos, como o método do gradiente descendente, a procura de Fibonacci, Golden Section e outros (LUENBERGER, 1973).

No entanto, se a função $f(X_i)$ for descontinua em vários pontos, possuir diversas restrições ou simplemente for suficientemente complexa, os métodos convencionais podem encontrar apenas máximos locais, mesmo depois de exaustiva procura. Neste caso, convém o emprego de um sistema genético de busca pelo máximo. Um algoritmo genético se propõe a criar indivíduos que representam soluções para $f(X_i)$, que neste caso representa a adaptação deste indivíduo ao meio. Um valor maior de função $f(X_i)$ representa uma melhor aptidão do indivíduo representado pela entrada X_i , o que deverá favorecê-lo no processo de evolução.

Este favorecimento será percebido na preferência pela reprodução, ou seja, os indivíduos que representam maiores valores de $f(X_i)$ terão preferência na passagem do seu código genético, fazendo com que haja tendência a convergência, ou seja, com que haja a cada nova geração uma inclinação a que novos indivíduos representem valores maiores de $f(X_i)$.

A maximização por algoritmos genéticos é definido pelos seguintes passos computacionais:

- 1. Criar população inicial aleatoriamente.
- 2. Calcular a adaptação de todos os indivíduos da população através da função $f(X_i)$ que se deseja maximizar.
- Criar descendentes através de reprodução, dando preferência aos que obtiverem maior adaptação.
- 4. Verificar critério de parada e retornar para o passo de número 2.

A reprodução (passo 3) pode ser dividida em seleção e a troca genética. A seleção terá o objetivo de encontrar aleatoriamente dois indivíduos na população, dando preferência àqueles com maior adaptação. Uma das possíveis técnicas a serem usadas é a "seleção por roleta" (DAVIS, 1991), em que um círculo, representando um jogo de "roleta", é dividido em N partes, onde N é o número de indivíduos na população.

As divisões da roleta deverão ter área proporcional à razão entre a adaptação de cada indivíduo e a soma de todas as adaptações. Assim, quando um número aleatório indicar a posição da roleta, a seleção será efetuada dando preferência a indivíduos cuja área da divisão seja maior, ou seja, os indivíduos mais aptos da população.

Após a seleção de diversos casais de indivíduos, a troca genética criará um ou mais descendentes por casal, que passarão a fazer parte de uma nova população. Esta troca genética pode ser efetuada pela média entre os pais, ou ainda por cruzamento, em que um ponto aleatório qualquer do código genético é escolhido e efetua-se o cruzamento, gerando dois descendentes, como demonstra a figura 5.9.

Os indivíduos resultantes da troca genética poderão também sofrer uma mutação, que de forma equivalente a inspiração biológica, modifica genes de alguns indivíduos de forma aleatória, podendo até mesmo torná-los mais aptos. A mutação é empregada com o objetivo de fazer com que o algoritmo genético não estacione em mínimos locais (DAVIS, 1991), favorecendo a busca pelo mínimo global.



Figura 5.9: Reprodução por cruzamento. Um ponto de cruzamento é escolhido aleatoriamente e dois filhos são criados

Deve-se guardar a resposta do melhor indivíduo de todas as gerações, pois este representa o máximo valor encontrado de $f(X_i)$. O critério de parada (passo 4) poderá ser um número máximo de gerações, um critério de tempo ou um limite para gerações onde nunca é encontrado valor superior ao melhor indivíduo encontrado.

Nesta dissertação os algoritmos genéticos foram empregados com o objetivo de auxiliar a criação de parâmetros para o critério automático de avaliação da posição da face, descrito na seção 7.2.2, e na obtenção de uma estimativa para o tamanho da face, descrita na seção 7.1.1.

5.4 Técnicas de processamento de imagens

Uma figura deverá passar por tratamentos matemáticos antes de estar apta a ser aplicada em uma rede neural. Algumas destas transformações, bem como a forma de representação de imagens, são mostradas nas subseções que se seguem.

5.4.1 Representação matricial de uma imagem e espaços de cores

A luz que provem de fontes naturais ou artificiais reflete em objetos e sensibiliza a retina humana, nos trazendo informações de forma e cor destes objetos.

A captação de uma imagem é feita de forma a reconhecer a luz que os objetos refletem, normalmente usando elementos sensíveis como o CCD (GROB, 1989). Quando esta captação é feita em conjunto com uma transformação para o domínio digital, para uso em computador, existe um parâmetro conhecido como resolução que deve ser levado em consideração. A luz refletida pelos objetos será representada por um número finito de pontos cujos valores de intensidade de luz serão armazenados em disco ou memória.

Estes pontos captados da imagem poderão representar novamente, caso requerido, em um monitor ou projetor, a mesma forma e cor que o objeto original possuía quando foi iluminado no momento da captação. Vale ressaltar que quanto maior for o número de pontos utilizados por centímetro quadrado de captação, melhor e mais fiel será a representação digital.

Para imagens que possuem apenas variações de níveis de cinza, cada ponto necessita armazenar apenas um valor, normalizado por exemplo entre 0 e 255, representando respectivamente o tom mais escuro e mais claro que um monitor ou projetor poderia emitir. Este valor tem o nome de luminância, e representa a luminosidade que o objeto possui em cada um dos pontos que o representam (GROB, 1989). A figura 5.10 mostra um exemplo de figura em tons de cinza, ou seja, composta apenas por luminância. Uma pequena parte da figura 5.10 tem seus valores de luminância mostrados em uma matriz numérica.

Já a imagem em cores possui três valores para cada ponto em sua representação matricial (GONZALEZ e WOODS, 1992), indicando uma mistura entre três bases de cores primarias, o que é conhecido como representação no espaço de cores.

Existem diferentes representações para os espaços de cores, sendo o espaço RGB (*Red-Green-Blue*) o mais comumente usado em armazenamento digital. Neste caso a luz refletida em cada ponto possui três valores indicando os níveis de vermelho, verde e azul também normalizados entre 0 e 255, representando os níveis máximo e mínimo de intensidade de cada cor. A figura 5.11 mostra um exemplo de figura usando o espaço de cores RGB, as matrizes numéricas mostram os valores de vermelho, verde e azul da região selecionada da foto.



Figura 5.10: Exemplo de figura em tons de cinza, a matriz numérica mostra os valores de luminância da pequena região selecionada.

Dessa forma, para representar uma imagem em aplicações computacionais utilizase normalmente uma representação matricial com valores de intensidade de cor. A matriz que contém a imagem terá o tamanho $X \times Y$ caso a imagem possua apenas variações de cinza ou apenas luminância, e $X \times Y \times 3$ caso a imagem seja em cores.

5.4.1.1 Espaço de cores RGB

O espaço de cores RGB é o mais conhecido e utilizado em muitas aplicações de armazenamento. Neste tipo de representação, cada ponto da imagem possui um valor para vermelho, outro para verde e outro para azul, simbolizando o que é chamado de mistura aditiva de cores, que também é usado em televisores e monitores (GROB, 1989).

Como é usado pela maioria dos modelos de captadores e câmeras (SIDNEY, 1991), o espaço de cores RGB normalmente é o mais indicado em aplicações em



Figura 5.11: Exemplo de figura colorida no espaço de cores RGB, as matrizes numéricas mostram os valores de vermelho, verde e azul da pequena região selecionada. (Em cores)

tempo real, onde pode não haver tempo para transformações. A figura 5.12 mostra as possíveis combinações entre vermelho, verde e azul em uma mistura aditiva de cores, formando outras cores além das primárias.

5.4.1.2 Espaço de cores YC_RC_B

A transmissão de televisão, bem como a maioria dos sistemas analógicos e digitais, utiliza a codificação YC_RC_B como método de representação de cores (GROB, 1989). Neste sistema, a cor de um ponto é representada por uma componente de luminância, representada por Y, e por duas componentes de cor $R - Y \in B - Y$.



Figura 5.12: Exemplo de misturas aditivas - vermelho, verde e azul são as cores primárias. (Em cores)

A compatibilidade com os sistemas preto e branco é uma das vantagens da utilização deste sistema de cores, pois na componente Y há a informação da figura em tons de cinza. Informações sobre a sensibilidade do sistema visual humano são usadas como parâmetros para a obtenção da luminância Y, que se refletem nas constantes (0,59) para verde, (0,30) para vermelho e (0,11) para azul. As equações em 5.10 definem este espaço de cores, a partir do espaço RGB convencional.

$$Y = 0,59G + 0,30R + 0,11B$$

 $C_R = R - Y$
 $C_B = B - Y$ (5.10)

5.4.1.3 Espaço de cores CIE XYZ

O espaço de cores RGB, descrito na seção 5.4.1.1, não é capaz de representar todas as cores reais possíveis, ficando limitado a pequena parcela, ajustada para corresponder à capacidade de fósforos de monitores e projetores.

O espaço de cores CIE XYZ foi criado para permitir a representação de qualquer cor, onde os estímulos X, Y e Z representam cores fora do espectro visível (PRATT, 2001). A figura 5.13 mostra um diagrama de cores contendo as variáveis X e Y do espaço de cores XYZ, a região em cores representa o espaço RGB, enquanto a região colocada em cinza representa a parte do espaço de cores que não pode ser obtida no espaço RGB.



Figura 5.13: Espaço de cores XYZ representado pelas variáveis X e Y. A parte em colorida representa o espaço RGB, enquanto a parte cinza representa o espaço não representável em RGB. (Em cores)

O conversão de RGB para XYZ pode ser obtida através da equação 5.11 representada na forma matricial. A conversão inversa, de XYZ para RGB pode ser efetuada pela matriz inversa da mostrada na equação 5.11. No entanto, se os valores de XYZ levarem a pontos na parte cinza da figura 5.13, os valores de R, G ou B correspondentes podem ser negativos ou maiores que a unidade, o que demonstra que nem todos os valores XYZ possuem correspondência em RGB. Assim, caso a conversão para RGB seja necessária, um prévio ajuste conhecido como Gamut (PRATT, 2001) deverá ser aplicado para permitir que os valores de R, G e B estejam entre 0 e 1.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0,49018626 & 0,30987954 & 0,19993420 \\ 0,17701522 & 0,81232418 & 0,01066060 \\ 0 & 0,01007720 & 0,98992280 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$
(5.11)

5.4.1.4 Espaço de cores CIE LAB

O espaço de cores CIE LAB separa uma imagem em três componentes, onde L representa a luminosidade ou brilho, e as componentes a e b representam parcelas de croma. O espaço LAB busca linearizar a perceptividade de cores, de forma a fazer as não linearidades das variáveis L, a e b reproduzirem a escala logarítmica do olho humano (PRATT, 2001).

A figura 5.14 mostra três gradientes de a e b com valores de L iguais a 30, 50 e 75% do máximo. Considera-se, neste caso, que os valores de L são representados entre 0 e 1 (100%), e que os valores de a e b estão representados entre -1 e 1.



Figura 5.14: Exemplo de gradientes de cores no espaço LAB, os valores de a e b representam parcelas de croma e os valores de L representam brilho. (Em cores)

A vantagem do sistema LAB é a que a representação de brilho da variável L está mais adaptada à forma com que o olho humano percebe a luminosidade. Por exemplo, uma modificação em torno de 18% na variável luminância Y do espaço de cores YC_RC_B transmitirá uma sensação de 50% menos brilho, o que significa que um ajuste linear na variável Y não resultará em uma modificação linear na imagem.

A conversão de RGB para LAB necessita de uma conversão prévia para o espaço de cores XYZ como mostrado na equação 5.11. As equações 5.12 e 5.13 mostram a

conversão de XYZ para o formato LAB, que possuem uma transformação cúbica representada pela função f(w) da equação 5.14 (PRATT, 2001).

$$L = \begin{cases} 116 \left(\frac{Y}{Y_0}\right)^{1/3} - 16 \quad \text{para} \quad \left(\frac{Y}{Y_0}\right) > 0.008856 \\ 903, 3 \left(\frac{Y}{Y_0}\right) \quad \text{para} \quad 0 \le \left(\frac{Y}{Y_0}\right) \le 0.008856 \end{cases}$$
(5.12)
$$a = 500 \left(f \left[\frac{X}{X_0}\right] - f \left[\frac{Y}{Y_0}\right]\right) \\ b = 200 \left(f \left[\frac{X}{X_0}\right] - f \left[\frac{Z}{Z_0}\right]\right) \end{cases}$$
(5.13)

$$f(w) = \begin{cases} w^{1/3} \quad para \quad w > 0.008856 \\ 7,787 \cdot w + 0,1379 \quad para \quad 0 \le w \le 0.008856 \end{cases}$$
(5.14)

Os valores de X_0 , Y_0 e Z_0 representam o branco padrão, escolhido como o D65 neste trabalho, ou seja, uma temperatura de cor de 6500K, muito usado em monitores e câmeras. Os valores para este branco são $X_0 = 0,950456, Y_0 = 1$ e $Z_0 = 1,088754$.

A transformação para LAB, em conjunto com as técnicas de equalização de histograma da seção 5.4.2, permitem uma correção e normalização da iluminação, representada pela variável L, sem modificar as componentes de cromaticidade, e levando em consideração as não linearidades da percepção do olho humano. Estas vantagens são empregadas, nesta dissertação, na correção de iluminação com o objetivo de obter melhor desempenho do sistema de detecção de pele da seção 6.1.1.

5.4.2 Equalização de Histograma

O Histograma de uma imagem é um gráfico da freqüência de aparecimento de cada faixa de tons de cinza. Quando o número de faixas de um histograma, também chamados "bins", é igual ao número de tons de cinza na imagem, o histograma representará uma contagem do número de vezes em que cada valor de tom de cinza foi encontrado na imagem. A figura 5.15 mostra um exemplo de cálculo de histograma com 256 "bins" em uma figura com 256 níveis de cinza.



Figura 5.15: Exemplo de cálculo de histograma com 256 "bins" em uma figura com 256 níveis de cinza.

A equalização de histograma é um método que acentua o contraste de uma imagem, ou seja, aumenta a diferença média entre os maiores e menores valores de luminância. A técnica de equalização de histograma é uma transformação nos valores dos tons de cinza da imagem de forma a permitir que o histograma cumulativo, também mostrado na figura 5.15, se torne uma função conhecida, como por exemplo, uma função linear (GONZALEZ e WOODS, 1992).

Após a aplicação da transformação, ou seja, quando o histograma cumulativo for representado por uma função linear, a freqüência de aparecimento de cada nível de cinza na imagem será a mesma, o que produzirá a equalização. O algoritmo de equalização de histograma pode ser representado pelos seguintes passos:

- 1. Calcular o histograma cumulativo cumhist(w) da imagem.
- 2. Multiplicar o histograma cumulativo por $255/(Tam_X \cdot Tam_Y)$ para efetuar uma normalização, onde Tam_x e Tam_y representam o tamanho da imagem.
- 3. Substituir cada valor de cinza w da imagem pelos valores normalizados de cumhist(w).

Após executado o algoritmo de equalização, o histograma cumulativo estará o mais linear possível, havendo apenas algumas não linearidades causadas pelos ar-

redondamentos do passo 2. A figura 5.16 mostra um exemplo de equalização de histograma, empregada nesta dissertação na detecção de pele descrita na seção 6.1.1 e na normalização das imagens do detector neural de faces da seção 6.3.



Figura 5.16: Exemplo de equalização de histograma, obtendo-se histograma cumulativo aproximadamente linear. A imagem original está na figura 5.15.

5.4.3 Técnicas de morfológicas de processamento de imagem

As técnicas morfológicas formam um conjunto de processamentos de imagem baseados na aplicação de modelos estruturais de vizinhança, onde o valor de cada ponto da imagem de saída será baseada em uma operação efetuada no mesmo ponto na imagem de entrada e em seus vizinhos dentro do elemento estruturante. As seções que se seguem descrevem as operações morfológicas de maioria, fechamento e preenchimento usadas nesta dissertação para compor o módulo de detecções de indícios de face, mais precisamente o filtro detector de pele, descrito na seção 6.1.1.

5.4.3.1 Filtro de maioria

No filtro de maioria o elemento estruturante é um quadrado com três pixels de lado, que é deslizado por sobre a imagem para permitir que os pontos de saída sejam decididos através da operação de maioria.

O filtro de maioria deve ser aplicado em imagens binárias, em que o valor de cada ponto pode ser "0" (preto) ou "1" (branco). A saída será "1" se a maioria dos

pontos dentro do elemento estruturante, que contém 9 pontos (3×3) for igual a 1, sendo "0" caso contrário.

A principal aplicação do filtro de maioria é remover ruído de uma imagem binária, normalmente representado por valores "0" ou "1" isolados, mantendo-se apenas os grandes blocos pretos ou brancos. Para garantir que todos os pontos representem a maioria dos vizinhos, o filtro de maioria deve ser executado quantas vezes for necessário até que a imagem de saída de uma execução seja igual a de entrada.

A figura 5.17 mostra um exemplo de aplicação do filtro de maioria, onde sobre a imagem original foi adicionado um ruído. Verifica-se que após a aplicação do filtro de maioria a imagem se assemelha bastante à original. Outros exemplos de aplicação do filtro de maioria podem ser encontrados na seção 6.1.1.



Figura 5.17: Exemplo de aplicação do filtro de maioria

5.4.3.2 Operação de fechamento

A operação de fechamento tem o objetivo de conectar áreas em branco separadas por pequenas áreas em preto, cuja separação for menor que a área do elemento estruturante.

O fechamento se baseia nas operações morfológicas de dilatação e erosão, executadas em seqüência e nesta ordem. A dilatação age deslocando o elemento estruturante, fazendo com que o ponto de saída seja "1" se pelo menos um dos pontos dentro do elemento estruturante de vizinhança for "1". A erosão age de forma inversa, fazendo com que cada ponto de saída seja "0" se pelo menos um dos pontos dentro do elemento estruturante for "0".

A figura 5.18 mostra um exemplo de operação de fechamento efetuada por uma dilatação seguida de erosão, onde o elemento estruturante é o quadrado branco mostrado na figura. No resultado final, mostrado à direita, a imagem original teve as áreas em preto fechadas por pontos brancos, pois a separação é menor que a área do elemento estruturante.



Figura 5.18: Exemplo de operação de fechamento efetuada por uma operação de dilatação seguida de uma erosão

O filtro de fechamento é empregado neste trabalho para conectar áreas de pele que estejam próximas, como descrito na seção 6.1.1.

5.4.4 Segmentação usando componentes conexos

O algoritmo de componentes conexos é uma técnica de segmentação aplicada a imagens binárias, cujo objetivo é separar áreas de pontos brancos conectados, o que, na prática, permite a separação e segmentação de objetos presentes na imagem.

Para efetuar a segmentação um procedimento procura pelo primeiro ponto branco presente na imagem, seguindo uma ordem da esquerda para a direita e de baixo para cima. Assim que o primeiro ponto é encontrado, uma etiqueta é colocada indicando que o mesmo pertence à área "A". A seguir, todos os 8 pontos imediatamente vizinhos ao primeiro ponto são checados, e todos estes que forem brancos também receberão a etiqueta "A".

Após a análise de vizinhança o algoritmo segue procurando novos pontos brancos. Quando são encontrados, uma nova etiqueta, por exemplo "B" é colocada. Durante a análise de vizinhança deste novo ponto, pode-se encontrar um vizinho em que tenha sido previamente colocada a etiqueta "A", o que significa que ambas as áreas estão conectadas. Neste momento o algoritmo unifica as duas etiquetas, ou seja, faz "A'' = "B''.

Após a execução completa do algoritmo, todas as áreas conectadas estarão separadas por diferentes etiquetas. Quando a vizinhança analisada contém todos as 8 direções possíveis, diz-se que a segmentação foi efetuada usando a conectividade D8. No entanto, o algoritmo de componentes conexos pode utilizar uma vizinhança com apenas 4 direções, para cima e para baixo, para a esquerda e para a direita. Neste caso diz-se que a segmentação foi efetuada usando conectividade D4.

A figura 5.19 mostra um exemplo de aplicação do algoritmo de componentes conexos. A imagem original possui 7 objetos brancos em um fundo preto, que foram segmentados através de 7 etiquetas diferentes, representadas pelas cores da imagem resultante à direita.



Imagem original



Imagem segmentada com componentes conexos

Figura 5.19: Exemplo de segmentação efetuada pelo algoritmo de componentes conexos. (Em cores) O algoritmo de segmentação por componentes conexos foi empregado nesta dissertação no procedimento de estimativa de face da seção 7.1.1.1 e durante a obtenção do banco de dados de tons de pele, descrita na seção 6.1.1.1.

5.4.5 Métodos de extração de bordas de uma imagem

Métodos usados para extrair bordas são muito úteis na segmentação de figuras (GONZALEZ e WOODS, 1992), constituindo uma importante técnica em possivelmente qualquer sistema de análise computacional de imagens.

Detectar bordas em uma imagem é encontrar a direção na qual existe uma grande variação do valor de nível de cinza. Alguns métodos de detecção de bordas, como o laplaciano (GONZALEZ e WOODS, 1992), envolvem a detecção dos pontos de zero da segunda derivada da imagem, pois nos pontos de grande variação de nível de cinza a primeira derivada será grande, possivelmente um máximo local, o que fará a segunda derivada ser zero.

Outra conhecida técnica é operador Sobel, que calcula o gradiente da imagem, que representa a direção na qual os níveis de cinza crescem mais rápido. A detecção destes pontos pode ser feita através de um limiar, ajustável de acordo com o tipo de imagem encontrada.

O cálculo do gradiente ou laplaciano normalmente envolve a convolução da imagem com uma determinada máscara, que representa a implementação discreta da função. A figura 5.20 mostra as máscaras para implementação de um operador Sobel, onde a implementação mais comum é o cálculo de $|G_X| + |G_Y|$ e de um laplaciano (GONZALEZ e WOODS, 1992).

$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$
Operad	or Sobel	Operador Laplaciano

Figura 5.20: Máscara para convolução que resulta no cálculo do operador Sobel (esquerda) e do laplaciano (direita) de uma imagem.

Os métodos Roberts e Prewitt (GONZALEZ e WOODS, 1992), também baseiam a detecção de bordas no cálculo do por gradiente, no entanto, empregam diferentes implementações para o cálculo de G_X e G_Y , como mostrado na figura 5.21.



Figura 5.21: Máscara para convolução que resulta no cálculo do operador Prewitt (esquerda) e Roberts (direita).

Um dos métodos considerados ótimos para detecção de bordas é o operador Canny (CANNY, 1986). Neste operador, inicialmente a imagem é filtrada por um operador gaussiano, como o mostrado na equação 5.15, com o objetivo de reduzir a interferência do ruído nas bordas detectadas.

$$\frac{1}{115} \begin{bmatrix}
2 & 4 & 5 & 4 & 2 \\
4 & 9 & 12 & 9 & 4 \\
5 & 12 & 15 & 12 & 5 \\
4 & 9 & 12 & 9 & 4 \\
2 & 4 & 5 & 4 & 2
\end{bmatrix} (5.15)$$

Depois da filtragem, O gradiente é calculado através da soma dos módulos de G_X e G_Y , obtidos com o uso de um operador Sobel, mostrado na direita da figura 5.20, e as bordas são obtidas através de um limiar, da mesma forma que em qualquer operador de gradiente.

Depois de obtidas as bordas, o operador Canny executa uma supressão de nãomáximos (CANNY, 1986), com o objetivo de afinar a borda encontrada, fazendo-a referir-se apenas ao máximo do gradiente.

Para evitar que as bordas contínuas sejam rompidas por problemas de ruído, o operador Canny também faz uso de dois limiares $T_1 \in T_2$, onde $T_2 > T1$. Como o valor do limiar T_2 é maior isto indica menor probabilidade de falsas detecções mas maior ocorrência de bordas quebradas, dessa forma o algoritmo busca nas detecções de limiar T_1 , bordas que sejam capazes de religar conexões nas bordas de limiar T_2 .

Estas modificações permitem ao operador Canny uma grande eficiência e imunidade ao ruído na detecção de bordas, mas em contrapartida necessita de maior tempo computacional. A figura 5.22 mostra um exemplo comparativo entre a detecção de bordas usando os operadores Sobel, Prewitt e Canny.



Figura 5.22: Exemplo de detecção de bordas por operador Sobel, Prewitt e Canny

A extração de bordas por operador Canny foi empregada nesta dissertação como método de localização indícios da presença face, descrito na seção 6.1.2. No entanto, a implementação do localizador de face em vídeo, descrito na seção 7.3, necessitou de um método mais rápido para detecção de bordas, optando-se por um operador Prewitt.

6 Contribuições desta dissertação

O objetivo desta pesquisa é adaptar e integrar diversos métodos usados para detectar tons de pele e faces, mostrados no estudo do estado da arte das seções 3 e 4, em apenas um sistema detector de faces cuja eficácia será verificada na seção 7. Para tanto novos métodos e técnicas foram criados, conforme se expõe nas sub-seções que se seguem.

6.1 Sistema localizador de indícios de face

O módulo segmentador, descrito na seção 6.2, divide a fotografia original em várias sub-imagens, para que seja possível testá-las separadamente com a rede neural da seção 6.3, especializada em detectar o padrão facial.

Entretanto, obter todas as sub-imagens possíveis de uma fotografia, permitindo sobreposição e sem qualquer tipo de limitação, certamente exigirá um grande número de passos computacionais, degradando o desempenho do sistema.

Para diminuir o espaço de busca, e por conseguinte o número de sub-imagens extraídas, propõe-se um sistema que verifique indícios da presença de uma face, selecionando regiões preferenciais onde haverá uma maior probabilidade de se encontrar faces.

Uma forma de obter indícios da presença de faces é fazer uso das informações de cor da imagem. O rosto humano certamente terá a cor característica de pele, o que pode ser detectado usando os métodos descritos na seção 3. A seção 6.1.1 propõe um sistema localizador de tons de pele usando redes neurais artificias.

Outro indicativo de face é a presença de bordas na imagem. Qualquer área da imagem que não possuir bordas muito possivelmente poderá ser excluída do espaço de busca, pois características faciais como olhos, boca e nariz geram bordas, como mostrado na seção 6.1.2.

6.1.1 Sistema localizador de tons de pele

Diversos sistemas detectores de tons de pele já foram propostos, como os descritos em (PEER e KOVAC et al., 2003), (JONES e REHG, 1998), (HSUAN YANG e AHUJA, 1998) e (BROWN e CRAW et al., 2001), previamente discutidos na seção 3.

Propõe-se, neste trabalho, localizar tons de pele usando redes neurais artificiais (RNA) (HAYKIN, 1994), sendo então necessário obter uma grande variedade de exemplos de tons de pele para efetuar a etapa de treinamento. Como se trata de um método ponto a ponto, descrito na seção 3.1, a RNA deverá então receber os valores de vermelho, verde e azul (RGB) de cada ponto da imagem e decidir, individualmente, se a mistura aditiva destes valores resulta ou não em um tom de pele.

6.1.1.1 Obtenção do banco de dados de tons de pele

Para constituir um banco de dados que possibilite o treinamento da RNA, obteve-se um grande número de imagens contendo pessoas através de motores de busca na Internet (GOOGLE, 2005) e (YAHOO!, 2005). Um método de extração de regiões foi criado para simplificar a catalogação dos tons de pele.

Inicialmente, o aplicativo desenvolvido requisita de um usuário que aponte, com o mouse, um ponto dentro de uma área onde esteja visualmente claro a presença de pele. Especifica-se, então, um limiar (α_{Limiar}) que representa a rigidez do algoritmo. O programa então aplica uma transformação em todos os pontos da imagem original seguindo o processo descrito na equação 6.1.

$$se R_{ponto selecionado} - \alpha_{Limiar} < R_{ponto} < R_{ponto selecionado} + \alpha_{Limiar}$$

$$e G_{ponto selecionado} - \alpha_{Limiar} < G_{ponto} < G_{ponto selecionado} + \alpha_{Limiar}$$

$$e B_{ponto selecionado} - \alpha_{Limiar} < B_{ponto} < B_{ponto selecionado} + \alpha_{Limiar}$$

$$então ponto = selecionado, senão ponto = não selecionado$$
(6.1)

Depois de executado o algoritmo representado na equação 6.1, obtêm-se todos os pontos da figura que estão na faixa definida por α_{Limiar} , ou seja, todos cujos valores RGB forem iguais aos do ponto selecionado pelo usuário ou na faixa entre $-\alpha_{Limiar} e + \alpha_{Limiar}$.

A seguir, um algoritmo de componentes conexos (JAIN, 1989), previamente descrito na seção 5.4.4, extrai toda a região de pontos selecionados pelo algoritmo da equação 6.1 que estão na região conexa iniciada pelo ponto indicado pelo usuário. A região extraída é mostrada ao usuário para validação final, sendo então integrado ao banco de dados de tons de pele. Três exemplos deste processo são mostrados na figura 6.1, o ponto inicial selecionado pelo usuário, neste caso, é o círculo na extremidade da linha azul, e o valor de α_{Limiar} usado foi de 10 pontos. Uma implementação em MATLAB para este segmentador manual de pontos de pele ou não-pele está documentada na rotina 16, no apêndice desta dissertação.



Figura 6.1: Três exemplos de extração de pontos de pele através da técnica de componentes conexos, aplicada dentro de um limiar de 10 pontos dos valores RGB. O círculo na extremidade da linha azul indica o ponto inicial selecionado pelo usuário. (Em cores)

O processo da figura 6.1 é então repetido inúmeras vezes, em muitas figuras diferentes, compondo-se então um banco de dados de exemplos de tons de pele.

Para efetuar o treinamento da RNA ainda há necessidade de obter exemplos de tons que não são pele, denominado tons de não-pele. Para isso, o mesmo algoritmo de seleção agora necessita que o usuário indique pontos em áreas das imagens que visualmente não representem pele. Exemplos de tons de cores das duas classes formadas são mostradas na tabela 6.1, os valores RGB estão entre 0 e 255.

6.1.1.2 Uso de rede neurais no localizador de tons de pele

Uma rede neural artificial (RNA) é então configurada com três entradas, para receber os valores de vermelho, verde e azul de cada ponto da imagem, e duas saídas, uma para indicar se o ponto representa um tom de pele e outra para indicar tom de não-pele. Esta rede neural é mostrada na figura 6.2.

Tons de Pele				Tons	de Não	o-Pele	
R	G	В	Exemplo	R	G	B	Exemplo
182	122	85		131	133	132	
201	149	99		238	238	214	
230	168	121		220	3	11	
178	118	66		82	19	62	

Tabela 6.1: Exemplos de pele e não-pele. (Em cores)



Figura 6.2: Configuração da rede neural usada para detectar pele. Não está representado o Bias. (Em cores)

Espera-se então, que os pontos de pele formem um agrupamento, ou cluster, separável do agrupamento dos pontos de não-pele com o uso de uma RNA. A figura 6.3 mostra uma representação gráfica dos pontos de pele de todo o banco de dados.

Como se percebe na figura 6.3, os pontos de pele formam um agrupamento cujo nível de vermelho é predominante sobre as componentes de verde e de azul. Na figura 6.4, o gráfico dos pontos de pele foi plotado separadamente nos eixos $Vermelho \times Verde$, $Vermelho \times Azul$ e $Verde \times Azul$. Desta forma, pode-se perceber que o valor médio do nível de verde é maior que o de azul, mas é menor que o nível de



Figura 6.3: Distribuição dos pontos de pele do banco de dados. (Em cores)

médio de vermelho. Os valores médios das distribuições são mostrados em cada um dos gráficos da figura 6.4 por marcadores azuis claro.



Figura 6.4: Distribuição dos pontos de pele separada em três gráficos que possuem os eixos $Vermelho \times Verde$, $Vermelho \times Azul$ e $Verde \times Azul$. (Em cores)

No entanto, o agrupamento dos pontos que não representam pele, ou seja, dos pontos de não-pele, mostrado na figura 6.5, engloba parcialmente o agrupamento dos pontos de pele. Isso ocorre pois não há qualquer impedimento da presença de objetos com a mesma tonalidade de pele presentes nas imagens que compõe o banco de dados. Trabalhos como (JONES e REHG, 1998) e (COTTON, 1996) também mostraram que a total separação dos pontos de pele e não-pele é bastante complexa.



Figura 6.5: Distribuição dos pontos de não-pele. (Em cores)

A figura 6.6 mostra os dois clusters no mesmo gráfico, onde vermelho indica ponto de pele e azul ponto de não-pele, e os pontos representados em magenta são os comuns aos dois clusters. Uma RNA então deverá ser treinada para separar o agrupamento azul do agrupamento vermelho, colocando os pontos em magenta no cluster mais provável, pois estes pertencem a ambas as classes.

Como a obtenção dos pontos de pele e não-pele é feita por semelhança no valor RGB, como descrito na seção 6.1.1.1, o banco de dados tem grande probabilidade de conter pontos repetidos. Em uma análise dos 39×10^6 pontos do banco de dados, verificou-se que apenas cerca de 552000 pontos são únicos, ou seja, mais de 98% dos pontos são repetidos.

Caso o banco de dados seja limitado a pontos que apareçam, no mínimo, 5 vezes, o número de pontos diminui para 166000. Este nível de repetição pode ser



Figura 6.6: Distribuição dos pontos de não-pele e pele mostrados no mesmo gráfico. Pontos de pele são mostrados em vermelho, pontos de não-pele são mostrados em vermelho e os pontos em comum aos dois bancos de dados são mostrados em magenta. (Em cores)

considerada um indício de confiabilidade deste subconjunto de 166000 exemplos, pois se um ponto não apareceu pelo menos 5 vezes em 39×10^6 provavelmente não é confiável considerá-lo representante de nenhum dos agrupamentos, pele ou não-pele. Propõe-se, dessa forma, considerar apenas este subconjunto diminuído do banco de dados durante o treinamento da rede neural artificial.

Para o conjunto de treinamento foi selecionado aleatoriamente do subconjunto diminuído do banco de dados cerca de 100000 pontos, mantendo-se a proporção 50% para pontos de pele e 50% para pontos de não-pele. Para o conjunto de validação, que será usado para parada antecipada do treinamento (HAYKIN, 1994), foram selecionados 15000 pontos também com proporções iguais.

O objetivo de usar proporções iguais nas classes no conjunto de treinamento é não causar na RNA uma maior tendência a classificar os pontos em uma das classes.

Todo os pontos não selecionados para treinamento ou validação são então incluídos no banco de dados de testes, que será usado para medir o desempenho do sistema. A tabela 6.2 mostra o treinamento de 10 diferentes RNAs, onde existe variação no número de neurônios da camada escondida (ZURADA, 1999), desde 2 neurônios na rede número 1, até 12 neurônios na rede de número 11. As redes neurais são completamente conectadas e a função de propagação (THOMÉ, 2005) usada foi a log-sigmóide, a mesma para todos os neurônios.

Tabela 6.2: Resultados de 10 redes para detecção de pele. Acertos (%) no subconjunto diminuído representa o desempenho em um banco de dados que possui apenas exemplos que se repetiram mais de 5 vezes. "Acertos (%) nos 39×10^6 " pontos representa o desempenho em todo o banco de dados.

Rede #	Configuração	Acertos (%) no subconjunto diminuído	Acertos (%) nos 39×10^6 pontos
1	3 2	88,38	$94,\!56$
2	4 2	89,15	94,88
3	5 2	89,54	95,00
4	6 2	89,69	$95,\!13$
5	7 2	88,88	94,22
6	8 2	89,02	94,24
7	92	90,42	$95,\!94$
8	10 2	90,26	$95,\!93$
9	11 2	90,19	95,82
10	12 2	90,12	94,12

Como se pode perceber nos resultados da tabela 6.2, a partir de 4 neurônios na camada escondida, a porcentagem de reconhecimentos corretos no banco de dados de testes se estabiliza perto de 90%. Este resultado pode ser considerado bom, pois como mostram as figuras 6.3, 6.5 e 6.6, os clusters de pele e não-pele são coincidentes em várias regiões nos gráficos.

Na tabela 6.2 é também apresentado o resultado do teste aplicado no banco de dados contendo todos os 39×10^6 pontos. Este teste pode ser considerado a melhor avaliação de desempenho do sistema, pois embora este banco de dados tenha muitas repetições, os pontos que mais se repetem são os mais importantes para serem acertados.

A tabela 6.3 mostra que outros quatro trabalhos obtiveram desempenho levemente inferior à proposta aqui apresentada, usando diversos outros métodos como clusterização 3d, modelos estatísticos e redes SOM. Isto mostra que a detecção de pele usando redes neurais obtém a mesma robustez de outras técnicas. Vale ressaltar que a comparação não foi feita sobre o mesmo banco de dados, já que nenhum dos trabalhos citados mostrou informações sobre quais testes foram utilizados, ou como foram realizados, apenas os resultados finais.

Tabela 6.3:	Comparação	deste	trabalho com	outros	métodos	de	detecção	de '	pele
-------------	------------	-------	--------------	--------	---------	----	----------	------	------

Trabalhog	Método utilizado	Reconhecimentos		
Trabamos	Metodo utilizado	corretos (%)		
Este trabalho	Redes MLP	95,94%		
(PEER e KOVAC et al., 2003)	Clusterização 3d	92,85%		
(JONES e REHG, 1998 $)$	Modelos estatísticos	88,00%		
(BROWN e CRAW et al., 2001)	Redes SOM	94,00%		

Alguns exemplos da aplicação do filtro de pele podem ser encontrados na figura 6.7. As figuras coloridas são do banco de dados FERET, melhor descrito na seção 7.2.1, e as em preto e branco representam a aplicação do detector, onde preto representa pele e branco não-pele. Em volta de todas as imagens em preto e branco da figura 6.7 foram colocadas bordas pretas para delimitar o espaço da imagem. Uma implementação em MATLAB (MATHWORKS, 2005) do filtro de pele está documentada na rotina 2, no apêndice desta dissertação.

A aplicação do sistema detector de pele pode ser bastante influenciada pela iluminação ambiente. No caso da fotografia no canto superior esquerdo da figura 6.7, a pouca iluminação e o fundo vermelho fez os pontos de pele tornarem-se muito escuros, não sendo detectados pelo algoritmo.

Uma forma de diminuir os efeitos de uma iluminação inadequada é aplicar uma equalização de histograma, como mostrado na seção 5.4.2. Para obter os efeitos desejados, a equalização deverá ser aplicada apenas na componente de iluminação, o que pode ser conseguido obtendo-se uma transformação no espaço de cores, que passa de RGB para CIE Lab. Dessa forma, as componentes de cor a e b são preservadas no processo. Detalhes sobre a conversão do espaço de cores RGB para o CIE Lab são mostrados na seção 5.4.1.4. Uma implementação em MATLAB (MATHWORKS, 2005) para a conversão de CIE Lab para RGB e vice-versa, bem como para a



Figura 6.7: Testes com o sistema detector de pele, imagens coloridas do banco de dados FERET e as preto e branco mostram a saída do detector, onde preto representa pele e branco não-pele. (Em cores)

correção de histograma propriamente dita, está documentado nas rotinas 9, 15 e 7 respectivamente, no apêndice desta dissertação.

A figura 6.8 mostra a aplicação do detector de pele com as mesmas fotografias usando-se previamente a equalização de histograma. As fotografias coloridas da figura 6.8 estão equalizadas, enquanto as em preto e branco representam a aplicação do detector de pele a partir da figura equalizada. Como se pode verificar na figura 6.8, a aplicação da equalização de histograma possibilitou uma melhor detecção na figura do canto esquerdo superior, onde havia iluminação inadequada.

As figuras 6.9 e 6.10 mostram outros exemplos de detecção de pele, aplicados em figuras que contém outros objetos além de faces.

UFRJ/IM/NCE



Figura 6.8: Testes com o sistema detector de pele onde é aplicado previamente uma equalização de histograma na componente de iluminação. Preto representa pele e branco não-pele. (Em cores)

A aplicação do detector de pele gera muitas vezes um conjunto de pontos isolados, constituindo um ruído na imagem detectada. Para retirar este ruído, e partindo-se do fato de que pontos de pele certamente terão muitos vizinhos, é aplicado um filtro de maioria, descrito na seção 5.4.3.1. Como este tipo de filtro somente mantém pontos iguais a maioria dos vizinhos, isto elimina pontos isolados na imagem. A figura 6.11 mostra a imagem de saída do detector de pele da figura 6.10 antes (esquerda) e depois(direita) da aplicação do filtro de maioria.

Normalmente os olhos, sobrancelhas e freqüentemente partes do nariz e boca não possuem cor de pele, portanto não serão detectados como tal. Para impedir que as sobrancelhas, olhos ou boca causem descontinuidades na detecção de indícios de



Figura 6.9: Detector de pele aplicado em figura contendo outros objetos além de faces. Preto representa pele e branco não-pele. (Em cores)



Figura 6.10: Detector de pele aplicado em figura contendo outros objetos além de faces. Preto representa pele e branco não-pele. (Em cores)

face é aplicada uma operação morfológica conhecida como fechamento, previamente descrita na seção 5.4.3.2.


Deteção de pele

Filtro de Maioria

Figura 6.11: Filtro de maioria (direita) aplicado depois do detector de pele (esquerda)

Após a aplicação do filtro, que utiliza um elemento estrutural quadrado de tamanho 40 pixels, todas as áreas de pele detectadas na figura que estiverem separadas por 40 pixels ou menos serão conectadas. Como este valor é menor que o tamanho das faces a serem detectadas, blocos de detecção de pele cuja separação for menor que a referida distância provavelmente constituem um só objeto, ou apenas uma face. A figura 6.12 mostra um exemplo de aplicação da operação de fechamento a partir do filtro de maioria.

Depois da aplicação do filtro de fechamento é aplicada uma operação de preenchimento de áreas que tornará pele todos os pontos que não forem identificados como tal mas estiverem cercados por pontos de pele. As operações de fechamento e preenchimento causam uma compactação dos indícios de face, o que permite que pontos principalmente no nariz, na parte superior da boca e inferior dos olhos, partes centrais da face, sejam considerados indícios mesmo que não possuam cor de pele, como se percebe na figura 6.12.

UFRJ/IM/NCE



Figura 6.12: Operações de fechamento, preenchimento e remoção de áreas menores que 5000 pixels

Uma última operação é necessária para melhorar o desempenho do sistema, e se encarrega de remover áreas de pele pequenas demais, mesmo após as operação de fechamento e preenchimento. Caso se saiba o limite mínimo do tamanho da face que se procura, áreas menores que, por exemplo, 50% do quadrado deste limite podem ser consideradas como erros, normalmente causados por objetos que possuem cor de pele e estão presentes na cena analisada. A figura 6.12 mostra uma eliminação de áreas menores que 5000 pixels, pois o tamanho mínimo de procura pela face foi considerado igual 100 pixels ($5000 = (100^2) \times 50\%$).

6.1.2 Sistema localizador de bordas

A informação de bordas em figuras foi utilizada para detectar faces em diversos trabalhos como (SAKAI e NAGAO et al., 1972), (GOVINDARAJU, 1996) e (HERPERS e KATTNER et al., 1995). Neste trabalho, a informação de bordas será usada da mesma forma que a detecção de pele, ou seja, apenas para diminuir o espaço de busca pela face, gerando indícios a serem usados pelo segmentador. Diversas técnicas de extração de bordas foram mostradas na seção 5.4.5.

6.1.2.1 Motivação para uso em detecção de faces

A busca pela face pode se tornar mais eficiente se regiões da figura original puderem ser excluídas do espaço de busca. Sabe-se que a presença de olhos, nariz e boca em qualquer face certamente gerará detalhes detectáveis por filtros de bordas. Regiões sem qualquer borda provavelmente compõem o fundo da figura e podem ser eliminadas do processo.

As figuras 5.22 e 4.1 das seções 5.4.5 e 4.2.1 respectivamente, mostram exemplos de como a detecção de bordas pode diminuir o espaço de busca. As fotos da figura 6.13 mostram que o fundo da imagem, onde não há bordas, poderá ser eliminado, tornando o algoritmo de detecção de faces mais rápido e possivelmente mais eficiente. O método de extração de bordas utilizado é um operador Canny (CANNY, 1986), previamente descrito na seção 5.4.5.

6.1.3 Obtenção dos indícios de face - pontos de borda-pele

Unindo-se a informação de pontos de pele com a de pontos de borda, obtém-se um bom indício da posição da face. No caso, a união é feita aplicando-se um operador E entre os pontos de borda e pele, estes novos pontos podem ser chamados de pontos de borda-pele.

Os pontos de borda-pele não necessariamente representam pontos de face, mas pode-se considerar que pontos que possuem cor de pele e contém bordas têm maior probabilidade de estarem em faces do que pontos que não têm cor de pele, ou não possuam bordas.

A figura 6.14 mostra o operador E aplicado entre os pontos de pele e bordas. Apenas os pontos brancos nas figuras da direita serão considerados pelo segmentador, diminuindo muito o espaço de busca pela face e o número de sub-imagens extraídas.

As figuras 6.16 e 6.15 também mostram os pontos de borda-pele extraídos de figuras com fundo mais complexo, onde existem outros objetos além da face. Uma implementação em MATLAB (MATHWORKS, 2005) do detector de bordas-pele está documentada na rotina 1, no apêndice desta dissertação.



Figura 6.13: Exemplos de detecção de bordas na diminuição do espaço de busca. As bordas detectadas estão representadas em preto. (Em cores)

6.2 Segmentador de sub-imagens

O módulo segmentador é responsável por cortar sub-imagens da figura original que está sendo testada pelo sistema localizador de faces. Para isso, deverão ser levados em consideração os pontos de borda-pele, descritos na seção 6.1.3, pois estes indicam locais onde a presença da face será mais provável.

A procura pela face deve ser feita dentre de limites de tamanho definidos pelo usuário, ou seja, será definido previamente os tamanhos mínimo e máximo da face procurada na figura.

O segmentador então deverá cortar, da figura original, todos os quadrados com tamanho de mínimo até máximo, centrados em todos os pontos de borda-pele en-



Figura 6.14: Exemplos de extração de pontos de borda-pele, representados em preto nas quatro figuras em teste. Os pontos de borda pele são extraídos através de uma operação E entre os pontos de pele e os pontos de bordas. (Em cores)

contrados, garantindo que todas as sub-imagens extraídas possuirão pelo menos um ponto de borda-pele. Neste caso, faz-se uma restrição para que todas as figuras extraídas sejam quadradas, o que, a princípio favorece que apenas faces quadradas sejam detectadas.

Segmentar apenas imagens quadradas torna o sistema detector de faces computacionalmente mais rápido, pois do contrário deveriam ser extraídas e testadas com o módulo neural um número muito maior de sub-imagens. Esta restrição está de acordo com a fisionomia da face, pois como mostra o histograma da razão altura



Figura 6.15: Exemplo de extração de pontos de borda-pele, representados em preto na figura em teste. (Em cores)

por largura de faces do banco de dados FERET, na figura 6.17, a maioria dos exemplos está contido no intervalo entre 0.9 e 1.1, o que mostra que faces quadradas ou próximas a isto são mais freqüentes.

Embora as faces detectadas pela rede neural sejam somente quadradas, durante a etapa de ajustes finos, descritos na seção 6.4.1, as faces detectadas poderão ser ajustadas para um retângulo, caso seja necessário.

Vale ressaltar que embora as sub-imagens obtidas pelo segmentador tenham tamanhos diferentes, isso será normalizado pelos métodos de processamento de imagens que são aplicados antes da apresentação destas à rede neural. Estes métodos de pré-processamento estão descritos na seção 6.3.2.

Entretanto, o número de imagens extraídas ainda poderá ser muito grande, dependendo do número de pontos de borda-pele. Foi criado então um método de redução do número de pontos, mantendo-se a sua característica e distribuição originais, mas possibilitando que o número de sub-imagens extraídas seja diminuído.



Figura 6.16: Exemplo de extração de pontos de borda-pele em uma figura com mais de uma face. Os pontos de borda-pele estão representados em preto. (Em cores)



Figura 6.17: Histograma Altura/Largura de 1935 faces do banco de dados FERET

6.2.1 Método de redução do número de pontos de borda-pele

Para diminuir o número de pontos de borda-pele, e por conseguinte o número de subimagens extraídas pelo módulo segmentador, o espaço contendo os pontos é dividido em um número n de partes iguais, tanto no eixo dos X, com divisões horizontais, como no eixo dos Y, com divisões verticais.

Para cada sub-divisão vertical, como mostra um exemplo na figura 6.18, um dos pontos no intervalo é escolhido aleatoriamente. Desta forma, quanto menor for o número de divisões n, menor será o número de pontos resultantes.

Por exemplo, considerando que a distribuição dos pontos de borda-pele está representada na figura 6.18. Se n = 6 então haverá seis divisões no eixo dos X como mostra a figura. Os pontos vermelhos foram escolhidos aleatoriamente dentro das faixas verticais mostradas na figura.



Figura 6.18: Exemplo de seleção de pontos de borda-pele no eixo dos X. (Em cores)

A mesma divisão em n partes é também realizada no eixo dos Y com divisões horizontais, como mostra a figura 6.19. Novos pontos serão escolhidos selecionados com seis divisões na vertical. Os pontos azuis da figura 6.19 foram escolhidos aleatoriamente dentro das faixas verticais.

Os pontos selecionados nas divisões verticais e horizontais constituem os pontos escolhidos para o módulo segmentador. O valor de n deverá ser indicado pelo usuário como uma precisão na busca pela face, e neste caso será mais apropriado indicar o número de pixels entre as divisões, pois como a figura original pode ter qualquer tamanho, o mesmo valor de n indicaria precisões diferentes a cada figura.



Figura 6.19: Exemplo de seleção de pontos de borda-pele no eixo dos Y. (Em cores)

A figura 6.20 mostra uma distribuição criada aleatoriamente com o objetivo de demonstrar o algoritmo utilizado, as distribuições mostradas na figura 6.21 mostram o algoritmo com divisões a cada 4, 3, 2 e 1 pixels.



Figura 6.20: Exemplo de uma distribuição de pontos criada artificialmente

Quando figuras com muitos objetos ou várias faces presentes são analisadas pelo sistema localizador de faces, freqüentemente os pontos de borda-pele formam vários



Figura 6.21: Exemplos do algoritmo de seleção de pontos aplicado em uma distribuição criada artificialmente, com intervalos iguais a 4, 3, 2 e 1 pixel

agrupamentos onde objetos com cor de pele ou faces podem estar presentes. Como a escolha dos pontos no algoritmo de redução é aleatória, pode haver uma preferência a algum agrupamento de pontos, causando erros de detecção.

Propõe-se, então, que os pontos de borda-pele sejam previamente clusterizados por um algoritmo k-means (JANG e SUN et al., 1997), descrito na seção 5.2. Sendo assim, a redução de pontos poderá ser melhor distribuída ao longo dos possíveis clusters.

O número de clusters será informado pelo usuário do sistema, devendo ser ajustado de acordo com a complexidade da foto a ser analisada, bem como o número de faces que se pretende encontrar. Ao término da aplicação do algoritmo K-means, os pontos de borda-pele estarão separados em clusters, sendo cada um deles analisado separadamente pelo algoritmo de redução mostrado anteriormente nesta seção. Para uma mesma precisão no algoritmo de redução, o aumento no número de clusters poderá indicar um possível aumento no número de pontos resultantes da redução, o que torna o sistema mais lento, sendo somente indicado em figuras contendo muitos objetos ou faces.

A figura 6.22 mostra um exemplo de imagem com duas faces onde foi aplicado o algoritmo de redução após uma clusterização com K-means. O número de clusters utilizado foi 3. O centro dos clusters está indicado por um círculo na imagem inferior esquerda da figura 6.22. Uma implementação em MATLAB (MATHWORKS, 2005) para a redução de pontos de borda-pele está documentada na rotina 12, no apêndice desta dissertação.



Figura 6.22: Exemplo do algoritmo de redução de pontos aplicado nos pontos de borda-pele clusterizados (número de clusters igual a 3). As cores vermelho, azul e roxo indicam os clusters, cujo centro está indicado por um círculo na imagem inferior esquerda. O algoritmo de redução é executado em cada cluster de forma independente. (Em cores)

UFRJ/IM/NCE

93

6.3 Detecção de faces com redes neurais

A detecção de faces usando redes neurais artificiais foi originalmente proposta em (ROWLEY e KANADE, 1999) e modificada nesta dissertação para incluir os métodos de indícios de face, descritos na seção 6.1. Foi também incluído um novo método de correção de iluminação, descrito na seção 6.3.2.1, além de um método de extração de características com o objetivo de tornar o sistema neural mais eficiente, como descrito na seção 6.3.4.

A base do processo de detecção usando redes neurais, assim como em (ROWLEY e KANADE, 1999), é o banco de dados de faces e não-faces, como mostra as seções que se seguem. Uma rede neural artificial modelo MLP (HAYKIN, 1994) será treinada com este banco de dados e terá capacidade de detectar se uma pequena subparte da imagem corresponde a uma face.

6.3.1 Banco de dados de faces e não-faces

O banco de dados de faces, originalmente construído por (ROWLEY e KANADE, 1999), é composto por um grande número de imagens de rostos em que olhos, nariz e boca foram centralizados com o objetivo de ter sua posição normalizada. A figura 6.23 mostra alguns exemplos deste banco de dados.

Além das figuras de faces, são recortadas também diversas imagens contendo ruídos, figuras geométricas aleatórias e sub-partes de imagens que foram erroneamente identificadas como faces pelo sistema usado por (ROWLEY e KANADE, 1999). Alguns destes exemplos estão expostos na figura 6.24

Tendo-se então os bancos de dado de faces e não-faces, pode-se treinar uma RNA capaz de detectar estes padrões. Ou seja, uma rede neural será treinada para receber as imagens de faces, na figura 6.23, e de não-faces na figura 6.24, e ter a capacidade de separar estas duas classes, de forma semelhante à rede neural que detecta tons de pele mostrada na seção 6.1.1.

No entanto, os mapas de bits que representam as imagens de faces e não-faces não podem ser apresentados à rede neural sem que haja um prévio tratamento, sob pena de comprometer a generalização da RNA. Várias etapas de pré-processamento foram propostas por (ROWLEY e KANADE, 1999) tais como: correção de iluminação, equalização de histograma, além de um marcaramento dos pontos laterais da imagem. Estas etapas estão descritas na seção 6.3.2.



Figura 6.23: Exemplos contidos no banco de dados de faces (ROWLEY e KANADE, 1999).

Outro processamento é necessário pois o formato de mapa de bits das imagens de faces e não-faces não é o mais adequado para apresentar a sistemas de reconhecimento de padrões usando redes neurais. Qualquer modificação na posição da face na figura, como por exemplo um leve movimento horizontal, vertical ou mesmo uma rotação certamente modificará todos os valores do mapa de bits, não mudando a classe da figura, ou seja, muito possivelmente não fará uma face tornar-se não-face. Um método de extração de características foi proposto nesta dissertação para tornar o sistema neural mais robusto quanto a estas modificações, sendo descrito na seção 6.3.4.

6.3.2 Métodos de pré-processamento das imagens do banco de dados de faces e não-faces

A normalização das imagens de face as torna mais semelhantes e, portanto, tende a favorecer a generalização da RNA, reduzindo variações de contraste e iluminação. Todos os métodos de pré-processamento são aplicados tanto nas figuras de faces



Figura 6.24: Exemplos contidos no banco de dados de não-faces (ROWLEY e KANADE, 1999).

quanto nas de não-faces, sendo o primeiro deles uma correção de iluminação. A seguir, na imagem com iluminação corrigida será aplicada uma equalização de histograma e finalmente a máscara de pontos laterais.

6.3.2.1 Correção de iluminação por aproximação linear

A iluminação presente nos objetos possivelmente modifica os valores de luminância de todos os pontos da imagem, mas não carrega nenhuma informação relevante à classificação. Se a iluminação for normalizada, será possível melhorar a generalização da RNA (ROWLEY e KANADE, 1999).

Partindo do princípio de que um objeto presente em uma imagem foi submetido a iluminação oriunda de apenas uma fonte de luz, podemos considerar que a iluminação pode ser representada por um gradiente linear somado à imagem, modificando os níveis de luminância. Dessa forma, se for possível estimar este gradiente, será possível subtraí-lo da imagem e obter uma figura iluminada por igual, ou seja, onde não haverá partes potencialmente mais claras que outras. A luz proveniente de uma fonte luminosa será atenuada de forma quadrática em relação à distância entre a fonte e o objeto a ser representado. Dessa forma, podese esperar que o gradiente somado à imagem seja também uma função quadrática, ou seja, se P é a potência em Watts, de uma fonte luminosa não direcional e r a distância em metros até o objeto, a distribuição de potência em Watts/ m^2 será dada por 6.2.

$$D(W/m^2) = \frac{P}{4\pi r^2}$$
(6.2)

Contudo, como as imagens de face são bastante pequenas, em torno de 19×19 pixels, o gradiente somado à imagem pode ser considerado linear (ROWLEY e KANADE, 1999), da mesma forma que partes pequenas de uma função quadrática podem ser representados por segmentos de reta.

Para estimar o gradiente, é proposto um método que calcula uma aproximação linear para uma função contendo as médias das colunas e linhas da imagem. As médias das colunas e linhas representarão, neste caso, a evolução da luminância média no sentido horizontal e vertical da imagem. Por exemplo, seja $\mu(x)$ um vetor contendo as médias dos valores de luminância das colunas da imagem:

$$\mu(x) = \{\mu_1, \mu_2, \mu_3, \mu_4, \dots \mu_n\}$$
(6.3)

Uma aproximação de $\mu(x)$ pode ser obtida através de um de uma regressão linear, como obtida através do método dos mínimos quadrados (MONTGOMERY e PECK et al., 2001). Assim, seja a função linear da equação 6.4.

$$\mu(x) \simeq z_x(w) = aw + b \tag{6.4}$$

Os valores dos coeficientes $a \in b$ podem ser calculados através das equações 6.5 e 6.6 (MONTGOMERY e PECK et al., 2001):

$$a = \frac{N_{elementos} \cdot \sum (\mu(x) \cdot w) - \sum \mu(x) \cdot \sum(w)}{N_{elementos} \cdot \sum (\mu(x))^2 - (\sum \mu(x))^2}$$
(6.5)

$$b = \frac{N_{elementos} \cdot (\sum w) \cdot (\sum \mu(x)^2) - (\sum \mu(x)) \cdot (\sum (\mu(x) \cdot w))}{N_{elementos} \cdot \sum (\mu(x))^2 - (\sum \mu(x))^2}$$
(6.6)

Onde $N_{elementos}$ é o número de elementos de $\mu(x)$ e w é um vetor dado pela expressão da equação 6.7:

$$w = \{1, 2, ..., N_{elementos} - 1, N_{elementos}\}$$
(6.7)

Considerando que o vetor $z_x(w)$ representa uma aproximação linear de $\mu(x)$, é possível calcular, através do mesmo método, $z_y(w)$, uma aproximação linear para $\mu(y)$, o vetor contendo a média de todas as linhas.

De posse das aproximações lineares $z_x(w)$ e $z_y(w)$, vetores contendo a média das colunas e linhas respectivamente, um produto externo destes vetores gerará um gradiente linear, que representa a parcela de iluminação aplicada à imagem. A figura 6.25 ilustra o processo, onde os gráficos em preto representam $\mu(x)$ e $\mu(y)$, e os gráficos em vermelho representam as aproximações lineares.



Figura 6.25: Sistema de correção de iluminação

A figura 6.26 mostra quatro exemplos em que a iluminação foi corrigida, são mostrados também os gradientes lineares calculados através do método dos mínimos quadrados. Uma implementação em MATLAB (MATHWORKS, 2005) para a correção de iluminação aqui proposta está documentada na rotina 3, no apêndice desta dissertação.



Figura 6.26: Quatro exemplos de correção de iluminação, incluindo o gradiente calculado

6.3.2.2 Equalização de histograma

A equalização de histograma tem o objetivo de normalizar o contraste das imagens, evitando imagens onde o contraste é muito reduzido, o que torna mais difícil a correta classificação da RNA.

O método aqui utilizado é o tradicional, que torna linear o histograma cumulativo da imagem, descrito na seção 5.4.2. A figura 6.27 mostra exemplos de histograma aplicados em imagens de faces. A equalização da figura 6.27 é feita a partir das imagens com iluminação corrigida de acordo com os métodos da seção 6.3.2.1.

Embora o tradicional método de correção de histograma esteja implementado no software MATLAB (MATHWORKS, 2005), uma implementação mais rápida e otimizada para figuras pequenas é possível, como documentado na rotina 8, no apêndice desta dissertação.

6.3.2.3 Filtro oval

Uma face em uma figura tem normalmente uma forma arredondada, portanto pixels nos cantos da imagem provavelmente não pertencem a face e assim não carregam informação relevante para a classificação, sendo possível retirá-los sem prejuízo à



Figura 6.27: Quatro exemplos de equalização de histograma, com as mesmas imagens de face da figura 6.26

detecção. O mascaramento, isto é, tornar pontos nas laterais iguais zero, impede que a rede neural faça uso dos valores presentes nos cantos da imagem durante a etapa de treinamento (ROWLEY e KANADE, 1999), favorecendo a generalização.

Propõe-se, então, a aplicação de uma máscara de pontos laterais, também chamada filtro oval. Em uma imagem pequena como as do banco de dados de faces e não faces, o efeito de uma máscara oval pode ser conseguido adicionando pequenos triângulos pretos à imagem, como mostra a figura 6.28.



Figura 6.28: Mascaramento dos pontos laterais

Os triângulos adicionados às laterais da imagem não devem ser demasiadamente grandes, sob o risco de mascarar os olhos ou a boca de uma face. Nas imagens do banco de dados de faces, que possui tamanho igual a 19×19 pixels, propõe-se usar

triângulos com 3 pixeis de lado, o que tornará preto 6 pixels por triângulo, 24 pixels por imagem de face, que possui 361 pixels.

A figura 6.29 mostra quatro exemplos de filtro oval aplicados sobre as imagens equalizadas da figura 6.27. Embora trivial, uma implementação do filtro oval em MATLAB (MATHWORKS, 2005) está documentada na rotina 11, no apêndice desta dissertação.



Figura 6.29: Mascaramento dos pontos laterais

6.3.3 Ordem de aplicação dos métodos de pré-processamento

A aplicação destas diferentes etapas de pré-processamento deve ser feita em uma ordem definida, de forma a obter-se a contribuição desejada de cada uma delas. Propõe-se que sejam aplicadas primeiramente a correção de iluminação, seguido da correção de histograma e finalmente do filtro oval.

Esta ordem de aplicação torna-se a única possível pois a correção de iluminação tende a retirar contraste da imagem com a subtração do gradiente, devendo ser aplicada imediatamente antes da correção de histograma. O filtro oval deve ser aplicado ao final do processo, como garantia de que os pontos laterais, que não carregam informação relevante, sejam realmente zero, de forma a impedir que estes valores sejam usados durante o processo de treinamento.

UFRJ/IM/NCE

O próximo passo antes do treinamento da rede neural é a extração de características como descrito na seção 6.3.4, seguido da aplicação do método de componentes principais (PCA), descrito na seção 6.3.4.1.

6.3.4 Métodos de extração de características e codificação da face

Mesmo após a aplicação do pré-processamento descrito na seção 6.3.2, as imagens de face e não face ainda necessitam ser codificadas para a apresentação à rede neural.

A mais simples codificação possível é usar os níveis de luminância do mapa de bits das imagens e apresentar à entrada da rede. Neste caso, cada um dos 361 (19×19) pixels que compõem cada imagem de face e não-face representam características a serem usadas pela rede neural durante o treinamento e testes.

Entretanto, uma pequena translação na posição de uma face dentro da imagem ou mesmo uma leve rotação modificará substancialmente todas as características apresentadas à rede neural artificial. Propõe-se, então, um método de codificação que extraia características das imagens de face e não-face tornando-as mais invariantes às modificações de posição e rotação. Este método de extração de características é conhecido como projeção poligonal, e está descrito na seção 6.3.4.3. Em ambos os casos as características extraídas são submetidas à aplicação de componentes principais, descrito na seção 6.3.4.1.

6.3.4.1 Aplicação de Componentes principais (PCA)

A técnica de componentes principais, previamente descrita na revisão teoria da seção 5.1.3, deve ser aplicada ao banco de dados de faces antes do treinamento da rede neural. O objetivo é reduzir a dimensão do vetor de dados e tornar as variáveis de entrada independentes, possibilitando com isso treinamentos mais rápidos e possivelmente uma melhor generalização (HAYKIN, 1994).

A aplicação de PCA exige duas etapas como mostra a seção 5.1.3. Na primeira, o banco de dados é analisado e os coeficientes para a projeção das novas variáveis são calculados, estes coeficientes formam a matriz de transformação. A primeira parte da aplicação de PCA utiliza apenas o banco de dados de faces, pois o banco de dados de não-faces possui apenas ruído e formas geométricas aleatórias, não contem informação relevante. A segunda parte do algoritmo de PCA é a projeção de novos dados no novo espaço de variáveis, o que é feito usando-se a matriz de transformação. Como a RNA necessita de vetores de entrada com tamanhos iguais, a segunda parte do algoritmo é feita tanto no banco de dados de faces quanto no de não-faces. Durante a operação do sistema de localização de faces, apenas a segunda parte do algoritmo PCA é executada, não havendo necessidade de se calcular a matriz de transformação novamente.

Dentro do novo espaço projetado através do algoritmo de PCA, pode-se retirar variáveis que possuam baixa variância, reduzindo a dimensão do vetor de entrada sem que se perca informação relevante. Para isso, durante a primeira fase do processo é indicado um valor em porcentagem que define o quanto se deseja perder de informação. Quando maior este valor, menor será o vetor de treinamento, sendo o valor ótimo definido experimentalmente.

6.3.4.2 Mapa de bits

Quando o método de extração de características é o mapa de bits, as imagens de faces e não-faces são pré-processadas, como mostrado na seção 6.3.2, e ocorre aplicação de PCA nas matrizes de bits das imagens, como mostra a seção 6.3.4.1.

Para testar a eficiência do mapa de bits, um conjunto de 32 redes neurais foi criado variando o número de neurônios da camada escondida e a porcentagem de aplicação de componentes principais. A figura 6.30 mostra um diagrama do modelo neural usado para detectar faces com mapa de bits.

O banco de dados foi divido em um conjunto de treinamento possuindo cerca de 10000 faces e não-faces, além de um conjunto de validação para parada antecipada contendo cerca de 3000 exemplos. Ambos os conjuntos foram criados contendo proporções iguais entre faces e não-faces.

No banco de dados de testes foram selecionados cerca de 2000 exemplos de faces e não-faces, em proporções iguais. O banco de dados de teste contém diversas faces ligeiramente rotacionadas e com posição não totalmente centralizada, como mostra a figura 6.31.

A dificuldade imposta à RNA no banco de dados de teste teve o intuito de verificar como se comporta a generalização quanto a estas variações, o que também foi abordado por (ROWLEY e BALUJA et al., 1998a).



Figura 6.30: Diagrama da rede neural usada para detectar faces com mapa de bits. Não está representado os Bias.



Sem pré-processamentos

Com pré-processamentos

Figura 6.31: Exemplos de faces do banco de dados de teste

A tabela 6.4 mostra as configurações das 32 redes neurais usadas neste teste e a porcentagem de PCA que foi aplicado no conjunto de amostras em cada teste. Por exemplo, a rede número 1 possui 40 neurônios na camada escondida e a aplicação de PCA é feita com limiar de 10^{-3} . Com isso, há uma perda de 0.1% ($10^{-3} \times 100\%$) de informação na projeção PCA, o que causa uma redução na dimensão da entrada

para a rede número 1, que passa de 361 (19×19) para 58. O número de saídas é sempre o mesmo, uma para indicar presença de face e outra para não-face.

#	Configuração da	% PCA	#	Configuração da	% PCA
	rede neural	,		rede neural	,
1	58-40-2	10^{-3}	17	302-80-2	10^{-5}
2	58-40-2	10^{-3}	18	302-80-2	10^{-5}
3	58-40-2	10^{-3}	19	302-80-2	10^{-5}
4	58-40-2	10^{-3}	20	302-80-2	10^{-5}
5	58-50-2	10^{-3}	21	302-90-2	10^{-5}
6	58-50-2	10^{-3}	22	302-90-2	10^{-5}
7	58-50-2	10^{-3}	23	302-90-2	10^{-5}
8	58-50-2	10^{-3}	24	302-90-2	10^{-5}
9	208-60-2	10^{-4}	25	302-100-2	10^{-6}
10	208-60-2	10^{-4}	26	302-100-2	10^{-6}
11	208-60-2	10^{-4}	27	302-100-2	10^{-6}
12	208-60-2	10^{-4}	28	302-100-2	10^{-6}
13	208-70-2	10^{-4}	29	302-110-2	10^{-6}
14	208-70-2	10^{-4}	30	302-110-2	10^{-6}
15	208-70-2	10^{-4}	31	302-110-2	10^{-6}
16	208-70-2	10^{-4}	32	302-110-2	10^{-6}

Tabela 6.4: Configurações de 32 redes para detecção de faces usando mapa de bits. Os resultados do treinamento estão na tabela 6.5

A função de propagação (THOMÉ, 2005) em todos os neurônios de todas as redes é tangente-hiperbólica à exceção dos neurônios de saída, onde foi usada a função sigmóide. A tabela 6.5 mostra os resultados das 32 redes no banco de dados de teste da figura da figura 6.31, onde a coluna "Acertos em faces" indica a porcentagem de acerto nas 1000 faces do banco de dados de testes e "Acertos em não-faces" indica a porcentagem de acerto nas 1000 não-faces.

De acordo com a tabela 6.5, a porcentagem de detecção das faces do banco de dados de teste fica em torno de 40% na maior partes das redes, enquanto no banco de dados de não-faces a porcentagem de acertos fica em torno de 98 %. Como existem apenas duas classes, pode-se dizer que o sistema classifica muitas faces da figura 6.31 como não-faces.

Tabela 6.5: Resultados do treinamento de 32 redes para detecção de faces usando mapa de bits. As colunas "acertos em faces" e "acertos em não-faces" representam a porcentagem de acerto no banco de dados de testes contendo faces e não-faces respectivamente. A configuração destas redes está na tabela 6.4

_//	Acertos em	Acertos em			Acertos em	Acertos em
#	faces $(\%)$	s (%) não-faces (%)			faces $(\%)$	não-faces (%)
1	39,9	97,2	17	7	40,3	98,3
2	30,9	96,6	18	3	25,7	96,6
3	31,2	96,6	19)	35,3	96,1
4	34,6	97,0	20)	38,1	96,5
5	42,7	97,8	21	-	29,4	96,4
6	34,2	96,4	22	2	$37,\!5$	94,2
7	33,8	96,3	23	3	35,7	95,6
8	35,7	96,7	24	Ł	36,6	96,5
9	35,3	95,9	25	5	31,8	96,2
10	34,4	96,3	26	5	35,9	96,4
11	36,8	96,0	27	7	$_{30,5}$	96,0
12	32,7	95,4	28	3	$_{30,5}$	96,5
13	28,3	97,0	29)	43,6	95,8
14	33,1	96,9	30)	39,2	96,5
15	34,9	96,2	31	-	38,8	96,6
16	35,1	95,7	32	2	37,9	93,6

6.3.4.3 Projeção poligonal de figuras em níveis de cinza

Ao invés de usar os valores de luminância do mapa de bits que formam as imagens de face e não-faces como características para o treinamento da rede neural, propõe-se um método de extração de características semelhante ao utilizado em (VIANNA e RODRIGUES et al., 2000) para caracteres manuscritos, chamado projeção poligonal.

Em (VIANNA e RODRIGUES et al., 2000), um caractere manuscrito de tamanho normalizado é colocado no centro de um polígono. A seguir são medidas as distâncias entre cada ponto que forma as arestas do polígono e o primeiro ponto que representa o caractere em teste. As distâncias extraídas são as características que serão usadas para o treinamento e teste de uma rede neural especializada em detectar caracteres. Contudo, este método não poderia ser utilizado para matrizes em níveis de cinza de faces e não-faces, por não haver como se arbitrar qual o primeiro ponto que representa a imagem. Para possibilitar o uso deste método, que obteve sucesso no aumento na generalização de sistemas de reconhecimento de caracteres (VIANNA e RODRIGUES et al., 2000), propõe-se um conceito de energia de projeção.

Para extrair as distâncias que compõe as características da projeção poligonal, uma figura de face ou não-face é então colocada em um polígono, que se propõe ser um quadrado como mostra a figura 6.32. A seguir arbitra-se um valor E_p para energia de projeção.



Figura 6.32: Exemplo de projeção poligonal usando um quadrado como base

Nesta modificação ao tradicional método de projeção poligonal (VIANNA e RO-DRIGUES et al., 2000), o valor da energia de projeção será subtraído dos valores de luminância da direção em que a distância estiver sendo medida. Por exemplo, a figura 6.33 mostra um conjunto de distâncias sendo medida em uma imagem, representada pelos valores de luminância. Se $E_p = 1$, então a energia de projeção será subtraída dos valores de luminância de cada uma das linhas até que seja menor ou igual a zero. No caso da figura 6.33, as medidas estão sendo tomadas a partir da aresta esquerda do quadrado, ou seja, uma para cada linha. Esta extração de medidas resulta nas distâncias (2, 3, 3, 3, 8, 3, 5, 6, 4, 3, 2), ou seja, o número de pixels na vertical desde a borda esquerda da figura até o ponto onde a energia é menor ou igual a zero.

Na figura 6.34 as medidas são tomadas a partir da aresta inferior do quadrado, ou seja, uma a cada coluna da imagem em teste. As medidas extraídas neste caso são (5, 2, 3, 4, 3, 5, 5, 6, 7, 4, 2, 5).

•											
L	-0,+	0,7	0,1	0,3	0,4	0,3	0,2	0,2	0,5	0,0	0,0
L	,.	-0,9 -	0,4	0,3	0,6	0,0	0,0	0,4	0,1	0,4	0,1
 	-0,0	-0,5	0,7	0,5	0,0	0,8	0,1	0,2	0,3	0,3	0,1
	-0,5	-0 , 1 -	0,4	0,7	0,2	0,5	0,4	0,6	0,0	0,1	0,5
с	0,0	0,0	0,0	0,1	0,0	0,5	0,2	0,3	0,2	0,6	0,2
	" ,T	0,8	0,4	0,2	0,3	0,5	0,8	0,4	0,2	0,0	0,1
	70,†	0,6	0,1	0,0	0,3	0,3	0,5	0,1	0,2	0,1	0,6
L	-0,+	0,2	6,0	0 ,3	-0,0 -	0,5	0,3	0,1	0,7	0,4	0,2
F	ס,ס	0,7	0,2	J,3	0,7	0,2	0,3	0,4	0,4	0,2	0,0
r	0,2	0,4	0,4	0,5	0,3	0,0	0,1	0,1	0,0	0,5	0,4
r	0,6	0,7	0,4	0,1	0,3	0,2	0,2	0,2	0,4	0,8	0,1

Aresta do quadrado

Figura 6.33: Exemplos de distâncias medidas a partir da aresta esquerda do quadrado. O conjunto de medidas extraídas é (2, 3, 3, 3, 3, 8, 3, 5, 6, 4, 3, 2). (Em cores)

0,4	0,7	0,1	0,3	0,4	0,3	0,2	0,2	0,5	0,0	0,0
0,9	0,0	0,4	0,3	0,6	0,0	0,0	0,4	0,1	0,4	0,1
0,0	0,3	0,7	0,5	0,0	0,8	0,1	0,2	0,3	0,3	0,1
0,5	0,1	0,4	0,7	0,2	0,5	0,4	0,6	0,0	0,1	0,5
0,0	0,0	0,0	0,1	0,0	0,5	0,2	0,3	0,2	0,6	0,2
0,1	0,8	0,4	0,2	0,3	0,5	0,8	0,4	0,2	0,0	0,1
0,1	0,6	0,1	0,0	0,3	0,3	0,5	0,1	0,2	0,1	06
01	0,2	0,0	03	0,0	0,5	03	C ,1	% 7	0,4	02
0 0	0,7	0,?	0 3	07	0,2	03	0,4	0,4	0,2	00
02	0,4	0,4	05	03	0,0	01	0,1	¢,0	C 5	04
06	0,7	0.4	01	03	02	02	0,2	6,4	08	01

Figura 6.34: Exemplos de distâncias medidas a partir da aresta inferior do quadrado O conjunto de medidas extraídas é (5, 2, 3, 4, 3, 5, 5, 6, 7, 4, 2, 5). (Em cores)

As distâncias serão também medidas a partir das arestas direita e superior do quadrado, compondo um conjunto de medidas que representará a figura original.

Como as imagens de face e não-face são quadradas com tamanho igual a 19×19 , cada aresta do quadrado conterá 19 medidas.

Durante a extração das medidas, deseja-se que características típicas de faces como olhos, nariz e boca bloqueiem a projeção, para que fiquem registradas nas distâncias. Sabendo-se que normalmente sobrancelhas, cílios e a borda dos olhos e boca são mais escuros que o restante da face, como se pode verificar nos exemplos da figura 6.35, pode-se fazer com que estas características faciais bloqueiem a projeção invertendo a imagem, ou seja, fazendo com que a luminância Y de cada ponto passe a 1 - Y.



Figura 6.35: Exemplos de como os olhos, nariz e boca são mais escuros que o resto da face, justificando que a imagem seja invertida antes de aplicar a projeção poligonal.

O processo de projeção poligonal e o conceito de energia de projeção aqui descritos se assemelham à forma com que são extraídas características de um objeto usando raios X, onde um objeto é colocado à frente de um anteparo e uma emissão de raios X sensibiliza um filme. O filme será mais ou menos sensibilizado de acordo com as características bloqueadoras do objeto. No caso da projeção poligonal, com a inversão da matriz da imagem, a "sensibilização" será ocasionada por regiões onde existem valores baixos de luminância.

Para evitar que seqüências de valores baixos na matriz da imagem, que com a inversão representam áreas brancas, diminuam demasiadamente a energia de projeção, propõe-se que a imagem já invertida tenha também seus elementos elevados ao quadrado. Com esta modificação, valores maiores que 0.5 na imagem original, que resultarão em valores menores que 0.5 com a inversão, não serão grandes o suficiente para atenuar a projeção mesmo que a energia inicial seja igual a "1". A equação 6.8 mostra a ordem das transformações sofridas por cada elemento y da matriz de imagem.

UFRJ/IM/NCE

109

$$y \to (1-y) \to (1-y)^2 \tag{6.8}$$

Em uma face possivelmente serão encontradas características mais ao centro da figura, para que estas não sejam perdidas no processo de projeção, que até agora foi feito de fora para dentro, propõe-se que sejam medidas distâncias de projeção do centro da figura para fora. Assim, a figura original é subdividida em quatro partes iguais, onde são retirados mais quatro conjuntos de medidas, como mostra a figura 6.36.



Figura 6.36: Projeção poligonal feita do centro para fora da figura, com o objetivo de registrar nas distâncias as características faciais que estejam eventualmente presentes no centro das imagens

As distâncias tomadas do centro para fora (figura 6.36) são medidas com o mesmo conceito de energia usado nas distâncias tomadas de fora para dentro (figura 6.36), incluindo as transformações de descritas na equação 6.8. No entanto, as medidas tomadas do centro para fora tem valor máximo menor, pois a projeção só pode passar por metade da figura. Para que a rede neural não leve em consideração esta diferença, todas as medidas são dividas pelo valor máximo possível, 19 nas projeções de fora para dentro e 9 ou 10 nas projeções tomadas do centro para fora, fazendo com que todas as distâncias extraídas estejam entre 0 e 1. Uma implementação em MATLAB (MATHWORKS, 2005) para a projeção poligonal aqui proposta está documentada na rotina 4, no apêndice desta dissertação.

UFRJ/IM/NCE

As figuras 6.37 e 6.38 mostram o primeiro e o segundo conjunto respectivamente de um exemplo de projeção poligonal. A imagem de face presente nas figuras 6.37 e 6.38 não está apresentada invertida ou elevada ao quadrado, mas pode-se perceber que os distâncias mostradas nos gráficos são menores quando o caminho da projeção possui pontos pretos, o que normalmente representa olhos, nariz e boca. As setas azuis indicam a direção onde são tomadas as distâncias



Figura 6.37: exemplo do primeiro conjunto (fora para dentro) de projeção poligonal de uma face

Nota-se que os triângulos do filtro oval, descrito na seção 6.3.2.3, foram colocados em y = 1 (branco), pois a inversão fará y = 0, valor que não bloqueará a projeção.

Na figura 6.37 pode-se perceber uma semelhança entre as medidas extraídas a partir da esquerda e a partir da direita da face. Esta semelhança é causada pela simetria presente em imagens de face, e como esta simetria está refletida nas medidas extraídas, a rede neural artificial pode, durante a etapa de treinamento, comparar as medidas e detectar faces usando a simetria como uma das características. Esta mesma semelhança pode ser encontrada no segundo conjunto de projeções, mostrado na figura 6.38, entre as medidas extraídas a partir do centro da face para a esquerda e para a direita.

As figuras 6.39 e 6.40 mostram outro exemplo de projeção poligonal aplicado a uma face, agora levemente rotacionada. Percebe-se que ainda existe semelhança en-



Figura 6.38: exemplo do segundo conjunto (centro para fora) de projeção poligonal de uma face

tre as medidas extraída pela direita e pela esquerda nos dois conjuntos de projeções, mostrando que a projeção poligonal pode permitir uma relativa invariância quanto a rotação.



Figura 6.39: exemplo do primeiro conjunto (fora para dentro) de projeção poligonal de uma face levemente rotacionada



Figura 6.40: exemplo do segundo conjunto (centro para fora) de projeção poligonal de uma face levemente rotacionada

Nas projeções superiores das duas faces, ou seja, nas projeções de cima para baixo das figuras 6.37 e 6.39 e do meio para cima das figuras 6.38 e 6.40, pode-se perceber que os olhos deixam uma clara diminuição na projeção, fato que uma vez retratado nos valores pode ser também usado no treinamento da rede neural. Já nas projeções inferiores, ou seja, nas projeções de baixo para cima e do meio para baixo pode-se perceber que a boca deixa uma marcação bloqueando a projeção no centro da imagem.

As figuras 6.41 e 6.42 mostram a projeção poligonal aplicada a um exemplo de não-face, que não possuí obviamente as simetrias e características encontradas nas projeções de uma face, o que tornará possível a separação entre os dois padrões durante o treinamento da rede neural.

As distâncias extraídas nos dois conjuntos de projeção, de fora para dentro (figura 6.32) e do centro para fora (figura 6.36), formam um vetor de características capaz de representar a imagem projetada. O tamanho deste vetor é dado pela equação 6.9, pois considerando que a imagem analisada é uma matriz quadrada $n \times n$, o número de distâncias extraídas será n para cada aresta do quadrado.

$$N_{\text{projeção poligonal}} = 2 \cdot 4 \cdot n = 8 \cdot n$$



Figura 6.41: exemplo do segundo conjunto (centro para fora) de projeção poligonal de um exemplo de não-face



Figura 6.42: exemplo do segundo conjunto (centro para fora) de projeção poligonal de um exemplo de não-face

$$N_{\text{mapa de bits}} = n^2$$

 $N_{\text{projeção poligonal}} < N_{\text{mapa de bits}} \text{ desde que } n > 8$ (6.9)

O tamanho do vetor de entrada em uma representação por mapa de bits é n^2 , o que indica uma diminuição no vetor de entrada com o uso de projeção poligonal desde que n > 8. No banco de dados utilizado neste trabalho tem-se n = 19, o que resulta em 361 características usando mapa de bits e 152 usando projeção poligonal.

Para efetuar a comparação entre a extração de características usando mapa de bits e usando projeção poligonal, foram treinadas 32 redes com a mesma configuração da tabela 6.4, usada no treinamento feito com mapa de bits mostrado na seção 6.3.4.2. A projeção poligonal é aplicada após o pré-processamento das imagens, descrito na seção 6.3.2 e antes da aplicação de PCA.

Nesta comparação entre as redes com características oriundas do mapa de bits e da projeção poligonal, o número de neurônios da camada escondida e o limiar de PCA foi mantido, além de ser utilizado o mesmo banco de dados de treinamento, teste e validação, como descritos na seção 6.3.4.2. No entanto, como a projeção poligonal resulta em um vetor de dados menor que o mapa de bits, como mostra a equação 6.9, o número de entradas das redes neurais será também menor. A tabela 6.6 mostra a configuração das redes usando projeção poligonal, e a tabela 6.7 mostra os resultados obtidos.

Uma comparação entre as tabelas 6.5, que mostra os resultados das redes com mapa de bits, e 6.7, que mostra os resultados com projeção poligonal, é mostrada na tabela 6.8. Nota-se que com o uso de projeção poligonal houve uma redução em torno de 23,5% no acerto de faces. Como o banco de dados usado nestes testes possui muitas faces levemente rotacionadas e com posição não centralizada, como mostrado na figura 6.31, isto mostra que o sistema de detecção usando projeção poligonal será mais sensível à presença de faces, adquirindo uma relativa robustez à rotação e translação. Entretanto haverá também uma maior tendência a falsas detecções, o que poderá ser evitado usando as técnicas de ajustes finos mostradas na seção 6.4.1.

6.4 Esquema final do Sistema localizador de faces

O sistema localizador de faces é a proposta principal desta dissertação, cujos blocos foram descritos nas seções 6.1, 6.2 e 6.3.

-#	Configuração da	% PCA	-44	Configuração da	% PCA
#	rede neural	70 I UA	#	rede neural	70 I UA
1	18-40-2	10^{-3}	17	150-80-2	10^{-5}
2	18-40-2	10^{-3}	18	150-80-2	10^{-5}
3	18-40-2	10^{-3}	19	150-80-2	10^{-5}
4	18-40-2	10^{-3}	20	150-80-2	10^{-5}
5	18-50-2	10^{-3}	21	150-90-2	10^{-5}
6	18-50-2	10^{-3}	22	150-90-2	10^{-5}
7	18-50-2	10^{-3}	23	150-90-2	10^{-5}
8	18-50-2	10^{-3}	24	150-90-2	10^{-5}
9	91-60-2	10^{-4}	25	152-100-2	10^{-6}
10	91-60-2	10^{-4}	26	152-100-2	10^{-6}
11	91-60-2	10^{-4}	27	152-100-2	10^{-6}
12	91-60-2	10^{-4}	28	152-100-2	10^{-6}
13	91-70-2	10^{-4}	29	152-110-2	10^{-6}
14	91-70-2	10^{-4}	30	152-110-2	10^{-6}
15	91-70-2	10^{-4}	31	152-110-2	10^{-6}
16	91-70-2	10^{-4}	32	152-110-2	10^{-6}

Tabela 6.6: Configurações de 32 redes neurais para detecção de faces com projeção poligonal. Os resultados estão na tabela 6.7.

O diagrama em blocos da figura 6.43 é uma ampliação do diagrama simplificado da figura 1.2, onde são detalhados os blocos segmentador, localizador de indícios de face e pré-processamento.

O protótipo do sistema localizador de faces foi implementado e desenvolvido em MATLAB (MATHWORKS, 2005). A seção 7.1 mostra mais detalhes sobre o protótipo incluindo uma avaliação de seu desempenho.

6.4.1 Ajustes finos no detector facial

A rede neural da figura 6.43 detectará quais sub-imagens cortadas pelo segmentador são consideradas faces. Como são cortadas figuras centradas nos pontos de bordapele que muitas vezes são próximos uns dos outros, é provável que uma mesma face na figura original seja detectada várias vezes em imagens adjacentes. Os ajustes finos evitam as múltiplas detecções e as empregam para remover detecções incorreTabela 6.7: Resultados do treinamento de 32 redes para detecção de faces usando projeção poligonal. As colunas "acertos em faces" e "acertos em não-faces" representam a porcentagem de acerto no banco de dados de testes contendo faces e não-faces respectivamente. A configuração destas redes está na tabela 6.6

_//	Acertos em	Acertos em	_//	Acertos em	Acertos em
#	faces $(\%)$	não-faces (%)	Ŧ	faces $(\%)$	não-faces (%)
1	$65,\! 6$	90,4	17	61,9	92,3
2	$55,\!6$	94,0	18	58,8	93,4
3	57,1	92,9	19	$54,\!5$	92,6
4	54,9	93,6	20	53,2	93,6
5	63,4	93,2	21	67,1	91,3
6	62,5	93,1	22	60,6	92,9
7	56,9	94,6	23	58,8	92,2
8	55,8	93,2	24	53,8	93,6
9	67,3	91,9	25	59,7	92,4
10	57,1	93,8	26	59,0	94,5
11	56,9	91,0	27	54,9	94,7
12	$58,\! 6$	92,1	28	60,3	93,4
13	66,9	92,1	29	64,9	92,3
14	$58,\! 6$	92,7	30	59,0	93,3
15	56,2	93,4	31	56,2	93,0
16	64,1	87,3	32	57,3	93,3

Tabela 6.8: tabela comparativa entre a extração de características por projeção poligonal e por mapa de bits

	Projeção poligonal							
	Acertos em faces $(\%)$	Acertos em não-faces $(\%)$						
Média das 32 redes	$59,\!3$	92,7						
Melhor rede	67,3	94,7						

	Mapa de bits							
	Acertos em faces $(\%)$	Acertos em não-faces $(\%)$						
Média das 32 redes	35,0	96,3						
Melhor rede	43,6	98,3						



Figura 6.43: Diagrama ampliado do sistema de detecção de faces

tas, sendo divididos em três etapas: o ajuste por espelhamento, descrito na seção 6.4.1.2, o ajuste de redução por centróide descrito na seção 6.4.1.1 e o ajuste por recomparação sucessiva, descrito na seção 6.4.1.3.

6.4.1.1 Redução por espelhamento

O espelhamento vertical é uma modificação imposta à imagem como se esta estivesse frente a um espelho colocado no eixo dos y, ou seja, faz-se uma mudança de variáveis onde os pontos das coordenadas x passam a $tam_x - x$, onde tam_x representa a largura da imagem. A figura 6.44 mostra um exemplo de imagem de face que sofreu um espelhamento vertical.

A simetria da face é a motivação para o espelhamento vertical, pois como mostrado na figura 6.44, embora a imagem depois do espelhamento seja diferente, também representa uma face.


Figura 6.44: Exemplo de espelhamento vertical de uma imagem. As faces, devido a simetria, não perderão suas características típicas.

Dessa forma, caso todas as faces detectadas sejam também espelhadas, os resultados também representarão faces. Caso o resultado da rede neural seja divergente, ou seja, se a rede neural considerar face uma imagem e seu espelhamento vertical for considerado não-face, isto muito possivelmente representará um erro, sendo eliminado do processo.

A figura 6.45 mostra um exemplo de redução por espelhamento, na imagem da direita o sistema detectou diversas faces, várias circulando a imagem de face propriamente dita, e algumas, mostradas em vermelho, que foram incorretamente detectadas como faces. Após o espelhamento, a rede neural não manteve a decisão de considerar as detecções em vermelho como face, o que indica um erro que não é passado às próximas etapas de ajuste fino. Uma implementação em MATLAB (MATHWORKS, 2005) para a redução por espelhamento está documentada na rotina 13, no apêndice desta dissertação.

6.4.1.2 Redução por centróide

A redução por centróide tem por objetivo principal evitar as múltiplas detecções, tendo-se apenas uma detecção para cada face na posição considerada mais provável. No sistema de detecção de faces em (ROWLEY e KANADE, 1999), os quadrados



Figura 6.45: Exemplo do ajuste fino de redução por espelhamento. Todas as detecções à esquerda sofreram espelhamento e foram novamente verificadas pela RNA. As detecções em vermelho não foram consideradas faces com o espelhamento, sendo removidas após o ajuste fino.

detectados cujos centróides forem vizinhos de D8 (JAIN, 1989) são reduzidos à média de seus vértices. Neste trabalho, propõe-se uma versão mais ampliada de vizinhança.

Quando o centróide de um quadrado B está sobre a área definida pelo quadrado A e vice-versa, define-se que os quadrados A e B são vizinhos de centróide. Este tipo de vizinhança foi empregada neste trabalho em substituição ao sistema usado em (ROWLEY e KANADE, 1999) pois a detecção de indícios de pele, mais precisamente a redução dos pontos de borda-pele (seção 6.2.1), faz com que quase não haja pontos vizinhos de D8.

Os quadrados vizinhos de centróide serão então reduzidos a apenas um, que estará na posição relativa à média de todos os vértices dos vizinhos. Porém, a média será ponderada de acordo com o valor presente na saída da rede neural que indica "face", ou seja, a saída da rede será usada como índice de confiança atribuído pela rede neural a cada detecção. A equação 6.10 resume a regra de redução por centróide onde $Q_{N=1,2,3...n}$ são os vetores contendo os vértices dos quadrados vizinhos de centróide, Q_f é o vetor com os vértices dos quadrados depois da redução e $C_{N=1,2,3...n}$ representam os índices de confiança.

$$Q_f = \frac{1}{n} \sum_{N=1}^{n} Q_n \cdot C_n$$
 (6.10)

Assim como a redução por espelhamento, a redução pode centróide é também usada para reduzir erros de detecção da rede neural. Aproveitando-se o fato de que as faces são muito freqüentemente detectadas várias vezes por quadrados vizinhos de centróide, quando um quadrado não possui vizinhos de centróide pode ser considerado um erro de detecção, não sendo levando em conta na média da 6.10.

A figura 6.46 mostra um exemplo de redução por centróide, a detecção representada pelo quadrado em vermelho não possui vizinhos de centróide e, portanto, não foi considerada na média.



Figura 6.46: Exemplo do ajuste fino de redução por centróide. Todos os quadrados amarelos são "vizinhos de centróide", tendo sido reduzidos a apenas um calculado através da média aritmética ponderada em relação ao índice de confiança. O quadrado em vermelho não foi incluído na média pois não possui "vizinhos de centróide". (Em cores)

6.4.1.3 Ajuste por recomparações sucessivas

Um último ajuste fino é uma busca com o objetivo de aumentar o índice de confiança indicado pela saída "face" da rede neural, fazendo pequenas modificações no tamanho e posição do quadrado detectado pela redução por centróide. Propõe-se, para tanto, uma busca gulosa visando maximizar o índice de confiança. Considera-se nesta busca que cada vértice do quadrado pode ser mantido, acrescido ou diminuído de um valor definido D a cada passo do algoritmo, a rede neural deverá checar 81 variações (3⁴) de tamanho a cada vez que um passo de busca for executado, mantendo-se o de melhor confiança.

O valor de D representa, neste caso, a precisão do processo chamado recomparação sucessiva. A busca local se encerra quando após um passo não há aumento no índice de confiança, assim o valor de D é diminuído pela metade e a busca se inicia novamente. Quando o valor de D for menor que 1 pixel o algoritmo faz a última busca local. Como a recomparação sucessiva pode adicionar ou reduzir valores diferentes a cada coordenada, há a possibilidade do resultado ser retangular, adaptando-se melhor à face.

A figura 6.47 mostra dois exemplos de recomparação sucessiva, onde o quadrado em vermelho é a entrada do algoritmo e o retângulo amarelo representa a saída. Uma implementação em MATLAB (MATHWORKS, 2005) para o ajuste por recomparações sucessivas está documentado na rotina 14, no apêndice desta dissertação.



Figura 6.47: Exemplo do ajuste por recomparações sucessivas. A marcação em vermelho representa a entrada do algoritmo e a marcação em amarelo a face melhor ajustada. (Em cores)

7 Desenvolvimento e Avaliação do Protótipo

Neste capítulo serão apresentados a metodologia dos testes, critérios e resultados do sistema detector de faces previamente descrito nas seções 6.3 e 6.4. Na seção 7.2 são propostos testes e a adoção de critérios automáticos de avaliação de desempenho usando o banco de dados FERET (FERET, 2005), com o objetivo de obter resultados sem a necessidade de inspeção visual. Na seção 7.3 a detecção de faces proposta é também avaliada sobre o vídeo obtido a partir de uma *WebCam* e finalmente na seção 7.4 sobre figuras com mais de uma face.

7.1 Protótipo do sistema detector de faces

O sistema proposto na seção 6.3, cujo diagrama em blocos é mostrado na figura 6.43, foi prototipado usando MATLAB (MATHWORKS, 2005) versão 7.0.0.19920 (R14), principalmente pela facilidade em testar aplicações que empregam imagens e redes neurais.

Através do MATLAB foi possível criar uma interface gráfica e acompanhar a evolução do algoritmo passo a passo, bem como executar o algoritmo em lote no banco de dados FERET. A figura 7.1 mostra o protótipo desta interface gráfica, onde ocorre a aplicação seqüencial das técnicas empregadas nesta dissertação para a detecção de faces.

No protótipo da figura 7.1, da esquerda para a direita e iniciando pelas figuras na parte superior, tem-se a aplicação do filtro de pele e de maioria (descrito na seção 6.1.1), a aplicação da detecção de bordas (descrito na seção 6.1.2), a localização dos pontos de borda-pele (seção 6.1.3) e a aplicação da técnica de redução de pontos de borda-pele (seção 6.2.1). Nas imagens na parte de inferior têm-se as faces detectadas pela rede neural artificial da seção 6.3 e a seqüência de ajustes finos descritos na seção 6.4.1, contendo a redução por espelhamento, por centróide e o ajuste por recomparação sucessiva.

7.1.1 Adaptações e ajustes ao banco de dados

O protótipo utilizado pode ser pré-ajustado para um tipo de banco de dados, de acordo com características próprias como a distância focal da câmera, qualidade da imagem, nível de ruído e outros parâmetros.



Figura 7.1: Protótipo do sistema automático de detecção de faces desenvolvido em MATLAB (MATHWORKS, 2005). São aplicados em seqüência, da esquerda para a direita; em cima: Filtro de pele, filtro de maioria, operação de fechamento, preenchimento e remoção de pequenas áreas, detector de bordas, localizador de borda-pele, redução dos pontos de borda-pele; em baixo: faces detectadas, redução por espelhamento, redução por centróide e recomparação sucessiva

O banco de dados FERET possui faces de tamanhos bastante variados, o que representa um desafio para o sistema detector de faces, pois o aumento da faixa

de procura pela face pode ocasionar o aparecimento de falsos positivos, como os mostrados em vermelho na figura 7.2. Verifica-se que o ajuste fino de redução por espelhamento, descrito na seção 6.4.1.1, eliminou um dos falsos positivos, mas manteve o outro.

A distribuição de tamanho de face no banco de dados FERET é mostrada na figura 7.3, onde cada face é representada por um "+" na posição que representa a sua largura (em x) e a sua altura (em y).



Figura 7.2: Falso-positivos em vermelho, detectados devido ao aumento da faixa de tamanho de procura pela face. Verifica-se que a redução por espelhamento eliminou um dos falso-positivos. (Em cores).

Conforme se percebe na figura 7.3, a distribuição dos pontos tende a uma reta em x = y, o que leva a conclusão que as faces frontais tendem a ser quadradas (altura=largura), o que já havia sido verificado no histograma da figura 6.17 na seção 6.2. Verifica-se também, que as faces estão distribuídas desde 100 × 100 até 300×300 , uma variação de 200%, que deverá ser coberta pela faixa de procura pela face, e possivelmente resultará em falsos positivos como os da figura 7.2, causando uma provável degradação no desempenho do sistema. Dessa forma, propõe-se que seja feito um ajuste para estimar o tamanho da face a partir da detecção de pele, descrito na seção que se segue.

7.1.1.1 Obtendo uma estimativa do tamanho da face

Para estimar o tamanho da face, possibilitando que esta seja encontrada mais rapidamente e que se minimize a detecção de falsos positivos, propõe-se um sistema que faça uso da detecção de pontos de pele, já empregada para detectar indícios da face e guiar o sistema neural de detecção.

Como a detecção de pele e os filtros de fechamento e preenchimento já fazem parte do sistema de detecção de indícios de face, descrito na seção 6.1.1, o processo



Figura 7.3: Distribuição do tamanho da face no banco de dados FERET. Cada face é representada por um "+" na posição que representa a sua largura (em x) e a sua altura (em y).

de estimativa de tamanho de face pode ser executado sem grandes prejuízos ao desempenho do sistema. Para tanto, considera-se que a maior área conexa da pele detectada na imagem corresponde à face.

Um exemplo do processo de estimativa do tamanho de face é mostrado na figura 7.4, onde a maior área conexa de pele é medida em número de pixels. O valor estimado para o tamanho de face $T_{Faceprob}$ é calculado através da raiz quadrada da área, ou seja, considera-se que a face é quadrada de área $(T_{Faceprob})^2$, como mostrado na equação 7.1.

$$T_{Faceprob} = \sqrt{Area_{Maior}} \tag{7.1}$$



Figura 7.4: Sistema de estimativa do tamanho da face. A partir da saída do filtro de preenchimento e fechamento, escolhe-se a maior área presente na figura e considerase a face. A raiz quadrada da área será considerada o lado da face. (Em cores)

Obviamente, como a face corresponde ao menor retângulo que possui olhos, nariz e boca, de acordo com a definição na seção 1.2, a estimativa será provavelmente maior que a face, contendo partes conectadas como pescoço, orelhas, queixo e outras, como se verifica na figura 7.4. Desta forma, propõe-se um sistema que calcule os valores de procura mínimo e máximo T_{INI} e T_{FIM} , respectivamente, baseado na estimativa $T_{Faceprob}$ e em coeficientes ajustáveis como α_{FIM} e α_{INI} da equação 7.2.

$$T_{INI} = T_{Faceprob} \cdot \alpha_{INI}$$
$$T_{FIM} = T_{Faceprob} \cdot \alpha_{FIM}$$
(7.2)

No entanto, a estimativa $T_{Faceprob}$ pode ser afetada por uma iluminação demasiadamente pequena ou oclusões na face causadas por barba ou óculos, tornando-a pequena demais por não cobrir a extensão da face. Também pode haver a influência de partes desnudas ou do fundo avermelhado de cor de próxima à pele, que poderia tornar a estimativa demasiadamente grande. A figura 7.5 mostra dois exemplos de cada erro, na parte superior a estimativa é pequena demais, causada por iluminação insuficiente. Na parte inferior, o fundo possui cor próxima à pele, causando erro na estimativa. A maior área conexa de pele detectada está representada em azul.

Para evitar estimativas demasiadamente grandes ou pequenas, como as da figura 7.5, propõe-se que existam limites máximos e mínimos, também ajustáveis. fazendo com que $T_{Faceprob}$ esteja obrigatoriamente em uma faixa definida, como mostrado na equação 7.3.

se
$$T_{Faceprob} < \beta_{MIN}$$
 então $T_{Faceprob} = \beta_{MIN}$
se $T_{Faceprob} > \beta_{MAX}$ então $T_{Faceprob} = \beta_{MAX}$ (7.3)

(7.4)

Os valores de α_{FIM} , α_{INI} , β_{MIN} e β_{MAX} devem ser ajustados para cada banco de dados de acordo com a distância focal ou modelo da câmera usada nos experimentos. Propõe-se que este ajuste, para o banco de dados FERET, seja efetuado com o auxilio de um algoritmo genético.

O algoritmo genético ajustará os valores das variáveis maximizando o número de vezes em que o sistema de estimativa gerar valores de T_{INI} e T_{FIM} que contenham, no intervalo, o tamanho correto da face. Para impedir soluções em que o tamanho de busca, ou seja, a diferença entre T_{FIM} e T_{INI} , seja muito grande, este é limitado em um valor determinado a cada experimento do sistema.



Figura 7.5: Exemplos de erro na estimativa do tamanho da face, onde a maior área conectada de pele, mostrada em azul, é considerada como face. Na parte superior a pouca iluminação gerou uma estimativa pequena demais. Na parte inferior o fundo com cor semelhante à pele gerou uma estimativa grande demais. (Em cores)

O gráfico na figura 7.6 mostra um conjunto de quatro experimentos executados com tamanho de busca limitado em 90, 95, 100 e 120 pixels. Nota-se que a % de acertos cresce conforme se aumenta o tamanho de busca, o que é esperado pois logicamente torna-se mais fácil estimar um valor dentro de uma faixa de tolerância maior. Como os testes preliminares mostraram que 91% das faces do banco de dados FERET estão entre 125 e 250 pixels, ou seja, um tamanho de busca de 125 pixels, pode-se verificar, através da figura 7.6, que um tamanho de busca de apenas 120 consegue-se, com o emprego da estimativa do tamanho de face, colocar dentro da faixa de procura cerca de 96% das imagens. Também se percebe, através da figura 7.6, que um aumento na faixa de busca além de 120 pixels não oferece ganho significativo.

UFRJ/IM/NCE



Figura 7.6: Tamanho de busca em pixels por % de acertos do algoritmo genético usado para ajustar as variáveis da estimativa do tamanho da face

As faces do segundo conjunto (segundo DVD-ROM) do FERET serão usadas pelo algoritmo genético para ajustar as variáveis do estimador de tamanho de face, deixando o primeiro conjunto (primeiro DVD-ROM) apenas para avaliação do desempenho do sistema. Em uma implementação comercial, propõe-se que esta fase seja executada como um ajuste fino do sistema, tornando-o mais rápido e confiável, adaptando-o à distância focal utilizada, iluminação ambiente e ruído presente na câmera, pois estes podem modificar a detecção de pele e por conseguinte a estimativa.

Os valores encontrados pelo algoritmo genético, ao fim de pelo menos 100 gerações, normalmente não diferem muito entre experimentos. Os resultados encontrados ficam em torno de $\alpha_{FIM} = 1.4616$, $\alpha_{INI} = 0,7522$, $\beta_{MIN} = 102$ e $\beta_{MAX} = 220$, o que significa que apenas uma estimativa entre 102 e 220 será considerada como válida, caso não esteja nesta faixa, será considerada 102 caso seja menor e 220 caso seja maior. Por exemplo, caso a estimativa seja 130, a procura será efetuada entre $T_{INI} = 98 (130 \times 0,7522)$ e $T_{FIM} = 190 (130 \times 1,4616)$. A tabela 7.1.1.1 mostra exemplos de cálculo de T_{INI} e T_{FIM} .

UFRJ/IM/NCE

Tabela 7.1: Exemplos do cálculo de T_{INI} e T_{FIM} , Faixa de procura pela definida pela estimativa do tamanho da face.

Estimativa	T_{INI}	T_{FIM}	Comentários
50	77	149	Estimativa menor que $\beta_{MIN} = 102$ - Foi ignorada,
100	77	149	$T_{INI} = \alpha_{INI} \times 102 \text{ e } T_{FIM} = \alpha_{FIM} \times 102$
120	90	175	
130	98	190	T_{INI} e T_{FIM} são calculados multiplicando a estimativa pelos valores
180	135	263	$\alpha_{FIM} = 1.4616 \ \alpha_{INI} = 0,7522$
200	150	292	
250	165	322	Estimativa maior que $\beta_{MAX} = 220$ - Foi ignorada,
450	165	322	$T_{INI} = \alpha_{INI} \times 220 \text{ e } T_{FIM} = \alpha_{FIM} \times 220$

7.2 Resultados usando o banco de dados FERET

O banco de dados FERET foi utilizado com base para a avaliação do sistema detector de faces, as seções que se seguem descrevem suas características e os critérios utilizados para avaliação automática do sistema.

7.2.1 Características do banco de dados FERET

O FERET (*"The Facial Recognition Technology"*) foi inicialmente um plano de estudo entre 1993 e 1997 patrocinado pelo "programa de desenvolvimento tecnológico antidrogas" da agência de defesa e pesquisa em projetos avançados dos Estados Unidos *"DARPA"*. O objetivo principal do projeto era desenvolver um sistema de reconhecimento de faces para apoio à inteligência e segurança para uso policial.

O banco de dados de faces criado para uso no projeto FERET está disponibilizado (FERET, 2005) desde 1998, tendo se tornado referência em diversos projetos na área. Estão contidas cerca de 14051 fotografias com uma pessoa apenas, de 1119 diferentes indivíduos em vários ângulos de pose e sujeitos a diversos tipos de iluminação e expressões faciais. A partir de 2003, o banco de dados FERET passou a disponibilizar imagens em cores, o que possibilita também avaliar o desempenho dos sistemas de detecção de indícios baseados em cores de pele.

Como o sistema descrito nesta dissertação foi proposto para uso em faces frontais coloridas, o banco de dados FERET fica reduzido a dois conjuntos, um com 1935

figuras contidas no primeiro DVD-ROM, e outro com 703 figuras no segundo DVD-ROM da distribuição. As imagens contêm apenas uma face cada e possuem 768 \times 512 pixels. A figura 7.7 mostra alguns exemplos das imagens presentes no banco de dados feret.



Figura 7.7: Exemplos de imagens presentes no banco de dados FERET. (Em cores)

7.2.2 Critérios automáticos de avaliação da posição da face

Para possibilitar uma avaliação automática, ou seja, sem que haja necessidade de inspeção visual, todas as 2638 faces do banco de dados FERET (conjunto do DVD 1 e 2) tiveram a face marcada manualmente, seguindo a definição da seção 1.2, ou seja, o menor retângulo que corta os olhos nariz e boca de um indivíduo. A figura 7.8 mostra mais quatro exemplos de faces do banco de dados, que tiveram a face marcada manualmente, representado pelo retângulo em azul.

A comparação entre a marcação feita pelo sistema automático proposto e a marcação manual será base para a avaliação do sistema. No entanto, haverá necessidade de um critério objetivo para comparação entre as duas marcações, pois na maioria das vezes haverá uma diferença entre estas.

Para definir este critério, propõe-se que sejam calculadas duas variáveis de avaliação. Uma das variáveis, denominada *acerto*, define a porcentagem de área comum entre a face marcada pelo sistema e a face marcada manualmente, ou seja, um *acerto* = 0.8 indica que a marcação do sistema cobre 80% da área da marcação manual.



Figura 7.8: Exemplos de imagens presentes no banco de dados FERET, com faces marcadas manualmente em azul. (Em cores)

Para que o sistema seja penalizado caso encontre faces muito maiores que a marcação manual, outra variável é definida como a porcentagem de área em excesso à marcação manual. Esta variável, denominada *amais*, também indica a qualidade da marcação, pois um valor alto de *amais* indica que a face detectada é muito maior que a marcada.

Por exemplo, caso uma marcação obtida pelo sistema seja avaliada com os valores de acerto = 0.9 e amais = 0.1, indica que 90% da área marcada manualmente foi coberta pelo sistema e houve um acréscimo de 10% da área marcada manualmente na área marcada pelo sistema.

A figura 7.9 mostra seis exemplos de marcação automática executada pelo sistema, mostrado em amarelo, comparados com a marcação manual, em azul. Os valores de *acerto* e *amais* são mostrados na parte superior de cada imagem do banco de dados FERET. O valor mostrado em amarelo acima da marcação automática da face representa o índice de confiança, descrito na seção 6.4.1.3.

Para obter uma avaliação que represente um critério visual, propõe-se que as marcações feitas pelo sistema e as feitas manualmente sejam submetidas a uma avaliação visual. Na figura 7.9, por exemplo, as marcações 1, 2, 3 e 6 são consideradas corretas enquanto as marcações 4 e 5 são consideradas incorretas, pois a área de acerto é pequena demais.



Figura 7.9: Exemplos do cálculo de *acerto* e *amais* em imagens do banco de dados FERET testadas pelo sistema. O cálculo das variáveis é feito comparando a marcação automática feita pelo sistema (em amarelo) e manual (em azul). O número na parte superior da detecção correta representa o índice de confiança, como descrito na seção 6.4.1.2. (Em cores)

Depois que as marcações de todos os faces do banco de dados FERET são avaliadas visualmente, propõe-se um sistema automático de validação que se baseia neste critério. A face será considerada correta se o valor de *acerto* for maior que um determinado limiar α_{acerto} e se o valor de *amais* for menor que um limiar β_{amais} . Os valores destes limiares, representados na equação 7.5, serão escolhidos de forma a maximizar a semelhança com a avaliação visual.

$$acerto \ge \alpha_{acerto}$$
$$amais \le \beta_{amais} \tag{7.5}$$

Propõe-se que a escolha de α_{acerto} e β_{amais} seja feita com o emprego de um algoritmo genético, onde cada indivíduo de uma população seja relacionado aos dois limiares, e a função objetivo seja minimizar o somatório dos erros quadráticos entre uma matriz de marcações feitas manualmente, e a executada com o critério da equação 7.5.

Após executado o algoritmo genético, a melhor solução, que possibilita uma semelhança de 96% com o critério visual, tem os valores de α_{acerto} e β_{amais} ajustados para 0,7 e 1,5 respectivamente, como mostrado na equação 7.6. Este passará a ser o critério automático utilizado para avaliar uma marcação de face feita sistema, ou seja, a face marcada automaticamente será considerada correta em relação a uma marcada manualmente caso a área comum seja maior que 70% e a área excedente não seja maior que 1,5 vezes a área da face marcada manualmente.

$$acerto \ge 0,7$$
$$amais \le 1,5 \tag{7.6}$$

7.2.3 Avaliação de desempenho usando o banco de dados FERET

A avaliação do sistema será executada usando-se o primeiro conjunto (primeiro DVD-ROM) do banco de dados FERET, visto que o segundo conjunto foi utilizado para ajustar os parâmetros de estimativa de face.

Cada teste completo, que procura faces em 1935 figuras de tamanho 768 \times 512 pixels, finaliza em torno 6 horas em um computador com processador Intel Pentium 4 com freqüência de *clock* de 2.3GHz, uma média de 16s por figura.

Foram executados 8 testes completos, onde diversos parâmetros do sistema foram ajustados buscando o melhor desempenho. A tabela 7.2 mostra as configurações dos testes executados, em que a avaliação do campo "% Acertos" foi efetuada segundo o critério automático da equação 7.6.

Número do Teste	1	2	3	4	5	6	7	8
T_{INI}	125	100	125	125	VAR	VAR	VAR	VAR
T_{FIM}	250	250	300	300	VAR	VAR	VAR	VAR
Passo	10	10	5	5	5	5	5	3
Largura da faixa de redução de pontos	3	3	2	2	2	2	1	1
Número de Faces na redução por centróide	10	10	10	20	10	20	20	20
Nível de recomparações Sucessivas	1	1	2	3	5	5	5	5
Redução por Espelhamento	não	não	não	não	não	sim	sim	sim
% Acertos	70,1	77,7	77,5	75,5	79,4	88,8	93,3	95,6

Tabela 7.2: Configuração dos diversos parâmetros do sistema e avaliação dos testes executados no banco de dados FERET

Na tabela 7.2, os valores de " T_{INI} ", " T_{FIM} " e "Passo" são, respectivamente, os tamanhos inicial, final e o passo incremental da faixa de procura pela face, que se torna variável a partir do teste 5, pois a partir deste empregou-se o sistema de estimativa de face descrito na seção 7.1.1.1. "Largura da faixa de redução de pontos" refere-se ao tamanho de cada divisão no algoritmo de redução de pontos de borda-pele, descrito na seção 6.2.1. O "número de faces na redução por centróide", descrita na seção 6.4.1.2, indica o número mínimo de vizinhos de centróide que uma detecção deverá possuir para não ser desconsiderada pelo sistema, enquanto o "nível de recomparações sucessivas" indica o número de pixels que será usado na busca gulosa do ajuste de recomparação sucessiva, descrito na seção 6.4.1.3. O ajuste fino de "Redução por Espelhamento" foi habilitado apenas após o teste 5, com o objetivo de reduzir o número de falsos positivos, como descrito na seção 6.4.1.1.

Através da avaliação de "% Acertos" na tabela 7.2, pode-se perceber que há uma melhoria no desempenho do sistema com a diminuição tanto da largura na redução de pontos como do passo incremental, que tornam o sistema mais preciso na busca pela face, mas em contrapartida ampliam o número de sub-imagens candidatas a face e por conseguinte causam um pequeno aumento no tempo de busca.

Com o aumento da precisão, uso de "Redução por Espelhamento", e desconsiderando detecções com menos de 20 vizinhos de centróide, obteve-se, no teste de número 8, um desempenho de 95,6%. A figura 7.10 mostra 8 exemplos de detecções corretas obtidas no teste 8, onde o número acima do retângulo que demarca a detecção é o índice de confiança atribuído pela rede neural, conforme exposto na seção 6.4.1.2.



Figura 7.10: 8 exemplos de detecções corretas feitas pelo sistema do teste 8 da tabela 7.2. O número acima do retângulo que demarca a detecção representa o índice de confiança atribuído pela rede neural, conforme exposto na seção 6.4.1.2. (Em cores)

A figura 7.11 mostra algumas detecções incorretas feitas pelo sistema do teste 8, causados principalmente por erros na estimativa do tamanho da face, problemas de iluminação e presença de óculos. Vale salientar que como os pontos de teste são escolhidos de forma aleatória pelo algoritmo de redução, uma repetição na busca pela face nas imagens da figura 7.11 muitas vezes resulta em acerto.



Figura 7.11: 8 exemplos de detecções incorretas feitas pelo sistema do teste 8 da tabela 7.2. O número acima do retângulo que demarca a detecção representa o índice de confiança atribuído pela rede neural, conforme exposto na seção 6.4.1.2. (Em cores)

Na seção 8.1 faz-se uma análise comparativa entre o desempenho do sistema proposto nesta dissertação e o reportado em vários trabalhos, usando-se o mesmo banco de dados FERET.

7.3 Resultados usando uma WebCam

Uma modificação no sistema proposto na seção 6.3 pode ser efetuada para ser aplicada em um sistema de vídeo, como o disponibilizado por uma câmera tipo WebCam. Neste caso, o sistema necessita ser rápido o suficiente para perder o mínimo de imagens (frames) de vídeo disponíveis durante a localização da face. Para isso, uma série de modificações são propostas.

No detector de pele, a equalização de histograma e os filtros de preenchimento, maioria e fechamento, descritos na seção 6.1.1, são desligados. Isto faz com que haja necessidade de uma boa iluminação, pois estes processos não estarão disponíveis para compensar eventuais variações.

Para a detecção dos pontos de borda-pele, a detecção de bordas é efetuada por um operador *prewitt*, que possui implementação mais rápida que o operador *canny*, embora possa não detectar bordas corretamente em uma imagem ruidosa (GONZALEZ e WOODS, 1992).

Usando uma faixa de tamanho desde faces a 50 centímetros até 1,5 metros de distância da câmera, as mudanças no sistema podem permitir que a localização de faces seja efetuada em aproximadamente 1s, em um computador com processador Intel Pentium 4 2.3GHz.

Após uma localização de face com sucesso em um frame disponibilizado pela câmera, o próximo frame não necessita ter todos os pontos de borda-pele novamente checados, a posição de face encontrada no frame anterior é verificada novamente em um frame atual para certificar-se que ainda é válida, o que pode ser efetuado em menos de 0,1s.

Para que pequenas variações na posição da face possam ser acompanhadas sem a necessidade de procurar faces *frame* a *frame* em toda a imagem, propõe-se que o ajuste fino de recomparação sucessiva, como descrito na seção 6.4.1.3 seja executado, procurando a melhor posição da face no frame atual a partir da posição encontrada no frame anterior. Caso a posição da face encontrada no *frame* anterior não seja mais considerada face no *frame* atual, a localização completa é novamente efetuada.

A figura 7.12 mostra um exemplo de acompanhamento da posição da face. No primeiro *frame*, na parte superior esquerda, toda o processo foi executado, enquanto nos *frames* subseqüentes apenas o ajuste de recomparação sucessiva foi suficiente para acompanhar as mudanças na posição da face.

A figura 7.13 mostra a resistência que possui o sistema a detectar faces rotacionadas. Nota-se que mesmo até cerca de 30° de rotação, para esquerda ou para a direita, a face continua sendo detectada.



Figura 7.12: Exemplos do acompanhamento da posição da face com o uso do ajuste de recomparação sucessiva. (Em cores)



Figura 7.13: Teste mostrando a resistência do sistema à rotação da face. (Em cores)

Variações de expressão também não influenciam a detecção de faces, como mostrado na 7.14, incluindo o caso onde os olhos estão fechados.



Figura 7.14: Exemplos de acompanhamento da posição da face com várias mudanças de expressão. (Em cores)

Outro exemplo de acompanhamento da posição da face é mostrado na figura 7.15. Neste caso houve um erro causado por um movimento brusco, seguido de uma detecção incorreta (figura central da parte inferior). No entanto, no frame subseqüente a face foi novamente detectada.

7.4 Resultados usando imagens com mais de uma face

A detecção de face em mais figuras contendo mais de uma pessoa não é tão confiável como a detecção de faces em vídeo (seção 7.3) ou em imagens do banco de dados FERET contendo apenas uma pessoa (seção 7.2).

Esta dificuldade é principalmente causada por não haver normalização quanto a posição ou tamanho da face, iluminação ou qualidade. Dessa forma, detectar faces sem qualquer restrição ainda representa um desafio computacional.

No entanto, se pelo menos uma estimativa do tamanho da face for definida pelo usuário, o sistema tem condições de apresentar resultados bastante expressivos.



Figura 7.15: Exemplos do acompanhamento da posição da face onde ocorreu um erro de detecção. (Em cores)

A figura 7.16, da seleção feminina de handebol, mostra um exemplo de detecção de múltiplas faces, em que também existem múltiplas escalas. A busca por faces foi efetuada desde 20 e 30×30 pixels de tamanho. No caso desta figura, a redução de pontos foi desligada, ou seja, todos os pontos de borda-pele são verificados. Isto foi feito pois a figura contém muitas faces e o tamanho da face é pequeno, qualquer ponto perdido pode significar uma face detectada a menos.

Pode-se notar na figura 7.16 que a maioria das faces foi detectada. Houve apenas um falso positivo completamente incorreto, ou seja, que não está na superfície da face, localizado na primeira fila, parte inferior, quarta jogadora. As faces na parte superior não puderam ser detectadas pois a baixa iluminação não permitiu que estes pontos serem incluídos como pele e posteriormente como borda-pele.

A figura 7.17, do time do Flamengo do Rio de Janeiro de 2001, mostra outro desafio ao sistema. Também nesta simulação foi desligado o sistema de redução de pontos. As faces estão também entre 20 e 30×30 pixels. Note que a iluminação ambiente é forte demais, tornando mais difícil a detecção de pele.

Adriano Martins Moutinho



Figura 7.16: Exemplo de detecção de múltiplas faces. Seleção feminina de handebol no pan-americano de 2003 (disponível em http://www.photoegrafia.com.br/). A maioria das faces foi encontrada. Um falso-positivo totalmente incorreto, na primeira fila, quarta jogadora. Detecção efetuada entre 20 e 30×30 pixels de tamanho. (Em cores)

Também desta vez, na figura 7.17, a maioria das faces foi detectada. Ocorreram dois falso-positivos completamente incorretos, nas pernas de dois jogadores, erro causado pela rede neural, pois estes pontos representam pele e possuem bordas. Note que em duas detecções incorretas a confiabilidade da detecção é baixa (valor mostrado em amarelo na parte superior da detecção), mostrando que o sistema pode estimar que a detecção possivelmente não resultou em face.

A figura 7.18 mostra um exemplo de grande complexidade, da foto de campeão do time do Flamengo do Rio de Janeiro de 2004. Há a presença de inúmeras faces, algumas em posição não frontal. A cor vermelha da camisa reflete este matiz para várias direções da foto, trazendo algumas dificuldades no detector de pele. Também



Figura 7.17: Exemplo de detecção de múltiplas faces. Time do Flamengo do Rio de Janeiro de 2001 (disponível em http://www.cmrevolution.com.br/). Iluminação forte demais. A maioria das faces foi encontrada. Dois falso-positivos completamente incorretos, nas pernas de dois jogadores. Detecção efetuada entre 20 e 30×30 pixels de tamanho. (Em cores)

desta vez não foi utilizada a redução de pontos e as faces foram procuradas entre 20 e 30×30 pixels.

O resultado da figura 7.18 pode ser considerado muito bom em face à dificuldade imposta ao sistema, quase todas as faces foram encontradas. Apenas quatro falsopositivos completamente incorretos foram encontrados, um na quarta criança da esquerda para a direita da parte inferior, outro na mão do penúltimo jogador que segura uma criança, e dois nas pernas das crianças da parte inferior esquerda da imagem.

UFRJ/IM/NCE



Figura 7.18: Exemplo de detecção de múltiplas faces. Time do Flamengo do Rio de Janeiro campeão Carioca de 2004 (disponível em http://lancenet.ig.com.br/). Grande complexidade e presença de várias faces, algumas não frontais. Quatro falso-positivos completamente incorretos. Detecção efetuada entre 20 e 30×30 pixels de tamanho. (Em cores)

A figura 7.19 mostra que o sistema não responde bem à oclusão facial, da foto de formatura do curso de Matemática da UESC (Universidade de São Carlos). Não foi possível detectar as faces dos formandos da fila de trás, pois além de quatro destes estarem ocultos a iluminação em todos está muito deficitária.

A figura 7.20 mostra um exemplo em que foi possível aplicar a redução de pontos, garantindo uma melhor eficiência computacional, graças ao fato de que as fotos possuem faces maiores. Neste caso, empregou-se a clusterização para impedir que a redução de pontos dê preferência a algum conjunto de pontos de borda-pele, como



Figura 7.19: Exemplo de detecção de múltiplas faces com oclusão facial. Foto da Formatura 2003.2 do Curso de Matemática da UESC (Universidade Estadual de Santa Cruz) (disponível em http://www.uesc.br/). Perdeu-se uma face por problemas de iluminação além das que estão parcialmente ocultas. Detecção efetuada entre 20 e 30×30 pixels de tamanho. (Em cores)

descrito na seção 6.2.1. O número de clusters desta simulação foi definido em 10, e as faces são procuradas desde 35 até 70×70 pixels.

O resultado da 7.20 pode ser considerado bom pois a análise da figura foi finalizada em torno de 6 segundos, em um Pentium 4 2.26GHz, enquanto a análise sem redução de pontos pode necessitar de até 1 minuto. Apenas um falso-positivo foi encontrado, possivelmente devido à camisa cor de pele.

Finalmente, a figura 7.21 mostra quatro exemplos de figuras onde a procura de face foi efetuada com a mesma configuração do exemplo da figura 7.20, ou seja, faces entre 35 até 70×70 pixels e 10 clusters na redução de pontos.



Figura 7.20: Exemplo de detecção de múltiplas faces. Adriano e seu pai Antônio. Nesta foto foi possível aplicar o algoritmo de redução de pontos. Um falso-positivo devido à cor de pele da camisa. Foi aplicada a redução de pontos, o número de clusters foi definido em 10, e as faces são procuradas desde 35 até 70 × 70 pixels. (Em cores)

O resultado do teste na figura 7.21 pode ser considerado bom, pois com a mesma configuração detectou-se diversos tamanhos, enquadramentos e posições de face. Dois falso-positivos encontrados, na figura superior esquerda na camisa que possui cor próxima à pele e na figura inferior direita na abertura do vestido da atriz. Apenas uma face não foi detectada.



Figura 7.21: Exemplo de detecção de múltiplas faces. Apenas dois falso-positivos encontrados e uma face não foi detectada. Foi aplicada a redução de pontos, o número de clusters foi definido em 10, e as faces são procuradas desde 35 até 70×70 pixels. (Em cores)

8 Conclusões Finais

Neste capítulo são apresentadas as principais conclusões sobre o funcionamento e desempenho do sistema detector de faces descrito nesta dissertação. É feita uma comparação com outros trabalhos na área, onde é efetuada uma análise crítica comparativa.

8.1 Conclusões sobre o sistema de localização de faces

A tabela 8.1 mostra uma comparação de desempenho entre o detector de faces proposto nesta dissertação e os outros detectores encontrados na literatura, que também efetuaram seus testes usando o banco de dados FERET. O desempenho de todos os trabalhos é bem próximo, mas a comparação não se faz justa, pois os critérios de detecção empregados na maioria dos trabalhos da tabela 8.1 não está totalmente descrita, ou são menos restritivos que os usados nesta dissertação.

Comparação de métodos	Técnica empregada	% de acertos		
Mátodo proposto posto dissorto eño	Redes neurais para	$95,\!6\%$		
Metodo proposto nesta dissertação	detecção de pele e faces			
$(MOCHADDAM \circ DENTIAND 1007)$	Autofaces (descrito na	07.0%		
(MOGHADDAM e FENTLAND, 1997)	seção 2.3.2)	97.070		
	Características extraídas			
(TAKÁCS e WECHSLER, 1995)	R, 1995) por uma filtro gaussiano e			
	heurísticas			
(HUANG e GUTTA et al., 1996)	Árvores de decisão	96%		
(POWLEV & PALILLA et al. 1008b)	Redes neurais para	07 80%		
(ItOWLET & DALUJA et al., 19960)	detecção de face	91,070		

Tabela 8.1: Comparação dos métodos Usando o banco de dados FERET.

Em (MOGHADDAM e PENTLAND, 1997) não está claro qual o critério para considerar a detecção como válida, mas a detecção efetuada inclui toda a cabeça do indivíduo, o que resultaria em um valor bem maior da variável *amais*, portanto um critério bem menos restritivo. A figura 8.1 mostra um exemplo de detecção feita por (MOGHADDAM e PENTLAND, 1997), em branco, em contraste com a mais refinada detecção proposta nesta dissertação.



Figura 8.1: Exemplo de detecção efetuada por (MOGHADDAM e PENTLAND, 1997) (tracejado), que possui um critério menos restritivo, e da detecção efetuada por esta dissertação (linha contínua amarela). (Em cores)

Em (TAKÁCS e WECHSLER, 1995) também não há uma boa definição sobre o critério utilizado, mas pode-se verificá-lo com a análise de duas de suas detecções mostradas na figura 8.2. A figura completa representa a detecção segundo (TAKÁCS e WECHSLER, 1995) e o retângulo amarelo uma detecção considerada correta pelo critério desta dissertação.

Nota-se na figura 8.2 que o trabalho de (TAKÁCS e WECHSLER, 1995) se limita a detectar toda a cabeça, e não apenas a face. Pelo critério de detecção usada por (TAKÁCS e WECHSLER, 1995), a cabeça detectada deve preencher pelo menos 80% da área "real" tomada como referência, embora não exista nenhuma definição para a medição desta referência, esta é certamente maior que duas vezes a definição de face desta dissertação, tornando o critério bem menos restritivo, e os resultados não tão confiáveis.

Em (HUANG e GUTTA et al., 1996), não parece existir preocupação com os limites laterais e inferiores do enquadramento da detecção, apenas com o limite superior, que deverá estar abaixo da linha do cabelo. No entanto, para uma detecção ser aceita, os olhos, a boca e o nariz devem estar no quadro de detecção.



Figura 8.2: Dois exemplos de detecção correta de faces segundo (TAKÁCS e WE-CHSLER, 1995). A figura completa representa a detecção do trabalho e em amarelo a detecção proposta por esta dissertação. (Em cores)

Se a detecção de faces não for restritiva, ou seja, limitar-se apenas a detectar toda a cabeça sem qualquer melhor posicionamento, não existirão critérios que garantam que apenas a face estará na área delimitada. Neste caso, os sistemas de reconhecimento, tanto os baseados na informação da imagem, como o método de Autofaces (seção 2.3.2), quanto os baseados em características extraídas pela transformada de Gabor, (seção 2.3.1), terão mais dificuldades para efetuar o processo de reconhecimento.

No caso dos trabalhos baseados em autofaces, a localização dos olhos deve ser efetuada para que seja possível o alinhamento e posterior projeção no espaço de autofaces (PENTLAND e MOGHADDAM et al., 1994). Se a área delimitada for muito maior que a face, como acontece em (HUANG e GUTTA et al., 1996), (TAKÁCS e WECHSLER, 1995) e (MOGHADDAM e PENTLAND, 1997), um método complexo de localização do olhos será necessário, talvez tão complexo como o que identifica faces.

No entanto, se o menor retângulo que envolve olhos, nariz e boca estiver disponível, métodos simples como os expostos em (MOUTINHO e THOMÉ, 2003) podem ser usados para detectar estes padrões, possibilitando o preciso alinhamento requerido pela técnica de autofaces.

O trabalho mais relevante na área de detecção de faces é (ROWLEY e KANADE, 1999), que também emprega redes neurais para efetuar a localização da face. No entanto, não é aplicado nenhum método para localizar indícios de face, apenas algumas heurísticas para diminuição do espaço de busca.

Os critérios de validação de face usados em (ROWLEY e KANADE, 1999) são tão restritivos quanto os usados nesta dissertação. No entanto, o teste em (ROWLEY e KANADE, 1999) foi feito em apenas um subconjunto de 1001 figuras do banco de dados FERET, não sendo informados quais critérios de separação foram utilizados.

Um grande problema no algoritmo proposto por (ROWLEY e KANADE, 1999) é que este não emprega uma redução significativa no espaço de busca. Como não há detecção de pele, as figuras são utilizadas em níveis de cinza, sendo necessário uma média de 250000 janelas para efetuar uma detecção em uma figura de 320×240 . Isto significa que a figura original será cortada em 250000 janelas para posterior pré-processamento e verificação com a rede neural artificial.

O algoritmo proposto nesta dissertação necessita apenas de 5000 janelas para cada figura que possui mais do que o dobro do tamanho (768×512), devido ao uso do sistema de detecção de indícios de face, que diminui o espaço de busca. Considerando que nas simulações efetuadas o tempo necessário para concluir a etapa de detecção de indícios é praticamente o mesmo que para concluir a etapa de detecção de faces propriamente dita, pode-se considerar o algoritmo aqui proposto equivalente ao tempo de 10000 janelas, ou seja, pelo menos 25 vezes mais rápido, tendo desempenho bastante semelhante, como já mostrado na tabela 8.1.

8.2 Conclusão sobre as contribuições desta dissertação

Algumas das técnicas e métodos aqui apresentados podem ser consideradas contribuições deixadas por esta dissertação, tendo possíveis aplicações mesmo não relativas a detecção de faces.

O sistema detector de pele, por exemplo, pode ser considerado individualmente uma contribuição. O seu bom desempenho, sendo capaz de separar corretamente 95% dos pontos de um banco de dados de mais de 39 milhões de pontos, o faz confiável para ser aplicado em outros sistemas na área de visão computacional, como detectores de movimento sensíveis à pele humana, ou mesmo sistemas de controle inteligente de conteúdo adulto, normalmente verificado por heurísticas a partir da detecção de pele (ZHENG e DAOUDI et al., 2000). Outra contribuição é a adaptação da técnica de projeção poligonal, que anteriormente estava disponível apenas para imagens binárias. Este método de extração de características, com as modificações propostas nesta dissertação, poderá ser aplicado a qualquer outro banco de dados de imagens em que a conversão para binário não seja possível, tendo possíveis aplicações mesmo não relacionadas a detecção de faces.

A implementação do sistema detector de faces em Matlab (MATHWORKS, 2005) também pode ser considerada uma contribuição desta pesquisa, pois muitas das funções nativas do Matlab foram modificadas para se tornarem computacionalmente mais rápidas. Estas rotinas modificadas possuem o sufixo *"fast"*, e podem ser encontradas no CD que acompanha este volume, à exceção da rotina 8 (*histeqfast.m*), que está também descrita no apêndice desta dissertação.

Dentro da área de detecção facial, o detector de indícios formado pelo detector de pele e bordas também pode ser considerado uma contribuição, pois a sua aplicação pode permitir a diminuição do espaço de busca pela face, e em ambientes de fundo controlado, até mesmo permitir a detecção da face. Trabalhos futuros que visem detectar faces não frontais ou rotacionadas podem empregar outros métodos para efetuar a detecção, mas utilizar o mesmo detector de indícios, pois as características deste permitem diminuir o espaço de busca mesmo em diferentes poses.

8.3 Considerações finais e trabalhos futuros

Esta dissertação propôs um sistema de detecção de faces que possui bom desempenho sobre o banco de dados FERET, cujas faces contém diversas variações de tamanho, posição, raça e expressão facial (FERET, 2005).

O bom desempenho do sistema, em conjunto com as restrições impostas à correta localização da face (variáveis *amais* e *acerto*), mostra que a detecção de faces aqui proposta está adaptada a qualquer processo de reconhecimento que necessite de prévio alinhamento, como Autofaces ou extração de características usando transformada de Gabor.

O funcionamento em vídeo, mesmo com a baixa qualidade de uma câmera tipo *webcam*, pode ser considerado bom, pois o sistema, com as modificações propostas na seção 7.3, possui capacidade de acompanhar pequenos movimentos da face, enquadrando-se na motivação de ser usado para possibilitar comunicações em sistemas de vídeo comprimido, que podem usar a localização da face para transmitir apenas esta parte do vídeo com qualidade, diminuindo a banda requerida para vídeo conferência.

Ainda sobre o funcionamento em vídeo, a validação da posição da face obtida em diversos *frames* consecutivos pode ser usada para garantir e aumentar a confiabilidade tanto da detecção de face quanto de um sistema de reconhecimento, mostrando que a detecção aqui proposta pode ser escalonável para qualquer nível de qualidade \times velocidade.

A seção 7.4 mostrou que o sistema de detecção de faces também possui capacidade de detectar múltiplas faces, embora ainda seja necessário algum tipo de estimativa do tamanho da face, sem a qual o sistema tende a encontrar alguns poucos falso-positivos.

Para concluir, o desafio de encontrar múltiplas faces em qualquer escala e pose ainda se encontra em aberto, no entanto este trabalho oferece uma opção computacional viável para a detecção de faces frontais, com leves rotações ou inclinações, em uma variação de escala relativamente grande.

8.4 Trabalhos futuros

Como trabalho futuro propõem-se melhorias na detecção de múltiplas faces, para que, por exemplo, o sistema torne-se capaz de estimar o tamanho da face a ser procurada, possibilitando assim o bom desempenho computacional necessário para uso em sistemas de catálogo, além diminuir o número de faces detectadas como falso-positivos.

Uma implementação futura em uma linguagem de alto desempenho também deve ser considerada, incluindo a possibilidade de efetuar parte do processamento em hardware, para tornar a detecção de faces mais rápida e confiável para aplicações em sistemas de vídeo em tempo real.

Outro trabalho futuro é verificar um possível ganho de desempenho para o sistema detector de faces com a implementação paralela dos sub-módulos do sistema. Sabe-se que uma rede neural MLP possui uma arquitetura que torna possível sua implementação paralela (HAYKIN, 1994), viabilizando certamente uma melhoria no desempenho tanto no detector de pele quanto na detecção de faces propriamente dita.
O tempo de resposta do sistema pode também ser diminuído caso outros submódulos sejam implementados em paralelo. Por exemplo, a detecção de indícios possui dois processos independentes, o filtro de bordas e de pele, que podem ser executados de forma simultânea em dois processadores separados.

Finalmente, o reconhecimento de faces constitui o mais importante trabalho futuro, cuja tarefa será encontrar a identidade da face previamente detectada pelo sistema aqui proposto. Em todos os sistemas de reconhecimento de faces pesquisados existe a preocupação, ou mesmo a necessidade, em delimitar corretamente a face antes que se extraia características que a representem, o que torna a correta detecção da face um aspecto primordial de qualquer sistema de reconhecimento de faces.

Apêndice de rotinas do MATLAB

Este apêndice destina-se a descrever as mais importantes rotinas que implementam os módulos do sistema detector de faces. Estas rotinas foram desenvolvidos em MATLAB versão 7.0.4.365 (R14) com Service Pack 2.

Os comentários, representados pelo delimitador %,estão em verde, e descrevem o funcionamento e sintaxe da rotina.

Índice de rotinas do MATLAB

1	checkbordapele.m
2	checkskin.m
3	corriglum.m
4	faceproject.m
5	fastcube.m
6	fastrootcube.m
7	histeqcolor.m
8	histeqfast.m
9	lab2rgb.m
10	metminquad.m
11	oval.m
12	pulapontosclust.m
13	reduzirnetflipretang.m
14	reduzirnetretang.m
15	rgb2lab.m
16	rgblabel.m
17	reshapeimg2net.m
18	tiraareas.m
19	variacoesmaismenos.m

Rotina 1: checkbordapele.m

1	function	mask =	checkbordapele ((netskin, img, lim, tamquadir	ni);
2					
0		1 . ~	1 • 1/ • /1 1	1) C	

 $3 \quad \% \text{ Executa a detecção de indícios (borda-pele) conforme}$

```
\% descrito na seção 6.1.
4
5
    \% mask = checkbordapele(netskin, img, lim, tamquadini);
6
7
    % mask é a imagem contendo os pontos de borda-pele em branco e todo
8
   % resto em preto. netskin é a rede neural que detecta pele. lim é o valor
    \%mínimo de saída da rede neural que se considera pele e tam<br/>quadini é o valor
9
10
    % da menor face que se espera detectar, para ser usado na remoção de pequenas
    % áreas.
11
12
13
    % pontos de pele
14
    entradaskin = checkskin (netskin, histeqcolor (img), lim);
15
16
    % filtro de maioria
17
    entradaskin = bwmorph(entradaskin, 'majority', 100);
18
19
    % filtro de fechamento
    entradaskin = inclose(entradaskin, ones(20, 20));
20
21
22
    \%filtro de preenchimento
23
    entradaskin = imfill(entradaskin, 'holes');
24
25
    % remoção de pequenas áreas
26
    entradaskin = tiraareas(entradaskin, 0.01 * tamquadini.^2);
27
28
    entradaedge = edge(rgb2gray(img)); \% bordas
29
30
    mask = entradaedge.*entradaskin; % borda-pele
```

Rotina 2: checkskin.m

```
function mask = checkskin(net, img, limiar);
1
2
3
    % Função que executa a detecção de pele (apenas aplicação da RNA).
4
5
     \% mask = checkskin(net,img,limiar);
    % mask contém a imagem de máscara que possui "1" para pele e "0" para não-pele
6
    \% net é a rede neural
7
    \% limiar o limiar da rede que é considerado pele (0.5 default)
8
    % img é a imagem RGB.
9
10
11
     \mathbf{x} = \operatorname{size}(\operatorname{img}, 1);
     y = size(img, 2);
12
     imgc = reshapeimg2net(img);
13
14
```

```
15
    saida = simfast(net, imgc);
16
    [m, mask] = max(saida);
17
18
19
    pos = find (m < limiar);
20
21
    mask(pos) = 0;
22
23
    mask = (mask = 1);
24
25
    mask = reshape(mask, x, y);
```

Rotina 3: corriglum.m

1 function saida = corriglum(entrada) $\mathbf{2}$ 3 % Corrige iluminação da entrada, assim como descrito % na seção 6.3.2.1. 4 5% saida = corriglum(entrada) 6 7 % entrada é a pequena imagem de face ou não-face (19×19) 8 % e saída é a entrada com correção de iluminação aplicada. 9 10lin = 1:19;11 mediacol = sum(entrada)/19; % Média das colunas1213% al e a2 são os coeficientes do método dos mínimos quadrados. 1415[a1, a2] = metminquad(mediacol);16mediacol = sum(entrada, 2) / 19; % Média das linhas 1718%b
1 e b2 são os coeficientes do método dos mínimos quadrados. 1920[b1, b2] = metminguad(mediacol);2122% Gera parte x do gradiente 23 $\mathbf{x} = \mathbf{lin} \ast \mathbf{a1} + \mathbf{a2};$ 2425% Gera parte y do gradiente 26y = [1:19] * b1+b2;2728% Gera o gradiente XY 29filtro = y * x;30

```
31 % Calcula o filtro
32 filtro = filtro/max(max(filtro));
33
34 % Inverte o filtro
35 filtro = abs(filtro - 1)*255;
36
37 % Efetua filtro + saída
38 saida = uint8(filtro + double(entrada));
```

Rotina 4: faceproject.m

```
1
     function saida = faceproject (entrada, potencia)
 2
 3
     % Executa uma projeção poligonal na imagem, conforme seção 6.3.4.3.
 4
 5
     \% saida = faceproject(entrada, potencia)
     % entrada é a pequena imagem de face ou não-face (19 \times 19)
 6
 7
     \% saida é uma matriz 8 \times 19 com os 8 conjuntos de distâncias
     % da projeção poligonal e potencia é a energia da projeção.
8
9
10
     % Inverte e eleva ao quadrado!
11
     saida = zeros(8, 19);
12
     \exp_{\text{entrada}} = \operatorname{realpow}(\operatorname{entrada} - 1, 2);
13
14
     % projecao esq para dir;
     \exp_{\text{entrada_s}} = \operatorname{cumsum}(\exp_{\text{entrada}}, 2);
15
16
     val = sum((exp_entrada_s < potencia), 2) + 1;
     \% Divide por 20 para normalizar a projecao de 0 a 1
17
18
     saida(1,:) = val/20;
19
20
     % projecao cima para baixo;
     \exp_{\text{entrada_s}} = \operatorname{cumsum}(\exp_{\text{entrada}}, 1);
21
22
     val = sum((exp_entrada_s' < potencia), 2) + 1;
23
     \% Divide por 20 para normalizar a projecao de 0 a 1
24
     saida(2,:) = val/20;
25
26
     \% dir para esq;
27
     \exp_{\text{entrada_flip}} = \operatorname{flip}(\exp_{\text{entrada}});
28
     \exp_{\text{entrada}_s} = \operatorname{cumsum}(\exp_{\text{entrada}_f});
29
     val = sum((exp_entrada_s < potencia), 2) + 1;
     \% Divide por 20 para normalizar a projecao de 0 a 1
30
     saida(3,:) = val/20;
31
32
33
     % baixo para cima;
```

```
34
     \exp_{\text{entrada}_{s}} = \operatorname{cumsum}(\operatorname{flipv}(\exp_{\text{entrada}}), 1);
35
     val = sum((exp_entrada_s' < potencia), 2) + 1;
36
     \% Divide por 20 para normalizar a projecao de 0 a 1
     saida(4,:) = val/20;
37
38
39
     % Segundo conjunto de projecoes
40
     meio_ver1 = exp_entrada(:, round(end/2):-1:1);
     meio_ver2 = exp_entrada(:, round(end/2)+1:end);
41
     meio_hor1 = exp_entrada_flip (round (end/2): -1:1,:)';
42
43
     meio_hor2 = exp_entrada_flip (round (end/2)+1:end,:);
44
45
     \% meio para esq;
46
     \exp_{\text{entrada_s}} = \operatorname{cumsum}(\operatorname{meio_ver1}, 2);
     val = sum((exp_entrada_s < potencia), 2) + 1;
47
48
     \%Divide por 11 para normalizar a projecao de 0 a 1
     saida (5,:) = val(end:-1:1)/11;
49
50
51
     % meio para dir;
52
     \exp_{\text{entrada}_{s}} = \operatorname{cumsum}(\operatorname{meio}_{\text{ver}2}, 2);
53
     val = sum((exp_entrada_s < potencia), 2) + 1;
     \% Divide por 11 para normalizar a projecao de 0 a 1
54
     saida (6,:) = val(end:-1:1)/10;
55
56
57
     % meio para cima;
     \exp_{\text{entrada_s}} = \operatorname{cumsum}(\operatorname{meio_hor1}, 2);
58
59
     val = sum((exp_entrada_s < potencia), 2) + 1;
60
     \% Divide por 11 para normalizar a projecao de 0 a 1
61
     saida (7,:) = val(end:-1:1)/11;
62
63
     % meio para baixo;
64
     \exp_{\text{entrada_s}} = \operatorname{cumsum}(\operatorname{meio_hor2}, 2);
65
     val = sum((exp_entrada_s < potencia), 2) + 1;
     \% Divide por 11 para normalizar a projecao de 0 a 1
66
     saida (8, :) = val(end: -1:1)/10;
67
```

Rotina 5: fastcube.m

1 function saida = fastcube(entrada);

2

3 % Implementa, de forma mais rápida, saida = entrada³

- 4
- 5 % saida = fastcube(entrada);
- $6 \quad \% \ {\rm saida} = {\rm matriz} \ {\rm de} \ {\rm saida}$
- 7 % entrada = matriz de entrada

 $\begin{array}{ll} 8 \\ 9 & v = ((0:32768)/32768).^{(3)}; \\ 10 & \text{entrada} = \text{round}(1+\text{entrada}*32768); \\ 11 & \text{saida} = \text{double}(v(\text{entrada})); \end{array}$

Rotina 6: fastrootcube.m

```
function saida = fastrootcube(entrada);
1
\mathbf{2}
    % Implementa, de forma mais rápida, saida = \sqrt[3]{entrada}
3
4
5
    \% saida = fastrootcube(entrada);
    \% saida = matriz de saída
6
7
    \% entrada = matriz de entrada
8
    v = ((0:32768)/32768).^{(1/3)};
9
10
    entrada = round(1+entrada*32768);
    saida = double(v(entrada));
11
```

Rotina 7: histeqcolor.m

function a = histeqcolor(image)1 $\mathbf{2}$ 3 % Executa a correção de histograma na componentes de luminosidade 4 % function a = histeqcolor(image) 5% a é a imagem de saída já corrigida e image é a entrada. 6 7a = rgb2lab(image); % converte para L*a*b* 8 9 10a(:,:,1) = 255*(a(:,:,1)/100);a(:,:,2) = (a(:,:,2)) + 128;11 a(:,:,3) = (a(:,:,3)) + 128;1213a = uint8(a);14% Executa a equalização em L (luminosidade) 1516a(:,:,1) = adapthisteq(a(:,:,1));17a = double(a);18a(:,:,1) = 100*(a(:,:,1)/255);1920a(:,:,2) = (a(:,:,2)) - 128; $\mathbf{a}\,(\,:\,,:\,,3\,)\ =\ (\,\mathbf{a}\,(\,:\,,:\,,3\,)\,\,)\,-128;$ 2122

a = lab2rgb(a); % Converte novamente para o espaço RGB

Rotina 8: histeqfast.m function saida = histeqfast(entrada) % Versão mais rápida (do a função do MATLAB histeq) de equalização

4 % de histograma. A imagem deve ter tamanho 19×19 , otimizado para

5 % equalizar as pequenas imagens de faces e não-faces.

7 % saida = histeqfast(entrada)

8 ~~%saida é imagem de saída com histograma corrigido e

9 % entrada é a imagem de entrada

11 nn = imhistc (entrada, 256, 1, 255) '; % nn é o histograma de a

12

13 cum = cumsum(nn); % Encontra histograma cumulativo 14

15 T = (cum-1)/(360); % Normaliza histograma

16 T(T < 0) = 0;

17

 $\frac{1}{2}$

3

6

10

18 saida = grayxform (a, T); % Aplica a equalização

Rotina 9: lab2rgb.m

```
1
    function [R, G, B] = lab2rgb(L, a, b)
2
    \% function [R, G, B] = Lab2RGB(L, a, b)
3
4
    % Transforma o espaço de cores de LAB para RGB,
    \% como mostrado na seção 5.4.1.4.
5
6
7
    % Lab2RGB recebe matrizes que representam L, a, and b no espaço CIELab
    % e os transforma em RGB. Originalmente escrito por Mark Ruzon.
8
9
10
    \% function [R, G, B] = Lab2RGB(L, a, b)
    % R,G e B são os valores RGB de saída e L, a e b são os valores LAB
11
12
    % de entrada.
13
14
    if (nargin = 1)
       b = L(:,:,3);
15
       a = L(:,:,2);
16
       L = L(:,:,1);
17
18
    end
```

```
19
20
    % limiares (recomendação ITU BT.709)
21
    T1 = 0.008856;
22
    T2 = 0.206893;
23
    [M, N] = size(L);
24
25
    s = M * N;
26
    L = reshape(L, 1, s);
27
    a = reshape(a, 1, s);
28
    b = reshape(b, 1, s);
29
30
   % Encontra o valor de Y
31
    fY = realpow(((L + 16) / 116), 3);
32
    YT = fY > T1;
33
    fY = (~YT) .* (L / 903.3) + YT .* fY;
34
    Y = fY;
35
36
    fY = YT \cdot (fastrootcube(fY)) + (~YT) \cdot (7.787 \cdot fY +
        16/116;
37
    fX = a / 500 + fY;
38
39
40
    fZ = fY - b / 200;
41
    fXZ = [fX; fZ];
42
43
44
    XZT = fXZ > T2;
45
    XZ = (XZT .* (realpow(fXZ,3))) + (~XZT) .* ((fXZ - 16/116) /
       7.787);
46
    MAT = \begin{bmatrix} 3.07993270842400 & -1.5371500000000 \end{bmatrix}
47
        -0.54278197539000
48
       -0.92123518073600
                             1.87599200000000
                                                  0.04524426122400
49
       0.05289097548800
                           -0.2040430000000
                                                  1.15115158049400;
50
51
    % Converter de XYZ para RGB
52
    RGB = MAT * [XZ(1,:); Y; XZ(2,:)];
53
54
    RGB(RGB < 0) = 0;
55
   RGB(RGB > 1) = 1;
56
57
    R = uint8(reshape(RGB(1,:), M, N) * 255);
    G = uint8 (reshape (RGB(2,:), M, N) * 255);
58
59
    B = uint8(reshape(RGB(3,:), M, N) * 255);
```

Rotina 10): metm	inquad.m
-----------	---------	----------

	1
1	<pre>function [m,b] = metminquad(entrada)</pre>
2	
3	% Implementa o método dos mínimos quadrados do vetor entrada
4	% que contém a média das linhas ou colunas de uma imagem.
5	% Usado para a correção de iluminação.
6	
$\overline{7}$	% function $[m,b] = metminquad(entrada)$
8	% m e b são os coeficientes da aproximação linear mx + b do método dos
9	% mínimos quadrados, e entrada é o vetor 1×19 contendo
10	% a média das linhas ou das colunas.
11	
12	somx = 190: $\%$ 1 + 2 + 3 + 4 + + 19
13	somx2 = 2470; $\% 1^2 + 2^2 + 3^2 + 4^2 + + 19^2$
14	som y = sum(entrada):
15	somxy = sum(entrada(1) + entrada(2)*2 + entrada(3)*3 +
16	$entrada(4)*4 + entrada(5)*5 + entrada(6)*6 + \dots$
17	$entrada(7)*7 + entrada(8)*8 + entrada(9)*9 + \dots$
18	entrada $(10)*10$ + entrada $(11)*11$ + entrada $(12)*12$
10	+
19	entrada(13)*13 + entrada(14)*14 + entrada(15)*15
10	
20	+ entrada (16) *16 $+$ entrada (17) *17 $+$ entrada (18) *18
20	
91	+ ontrada (10) *10).
21 99	$e_{1111}a_{11}a_{11}(19)*19),$
22 92	07 A entrode é encouris de non reu + h
20 04	7_0 A entrada e aproximada por mx + b
24	$\mathbf{b} = (\operatorname{som} x \operatorname{som} x 2 - \operatorname{som} x \operatorname{som} x y) / (19 \operatorname{som} x 2 - \operatorname{som} x \operatorname{som} x); \ \% \text{ Coefficiente } \mathbf{b}$
25	$\mathbf{m} = (19*\operatorname{somxy-somx*somy}) / (19*\operatorname{somx2-somx*somx}); \% \text{ Coefficiente } \mathbf{m}$

Rotina	11:	oval.m
--------	-----	--------

1 2	<pre>function [saida] = oval(entrada, level, cor)</pre>
$\frac{2}{3}$	% Executa o filtro oval, como descrito na seção 6.3.2.3
$\frac{4}{5}$	$\% \ [\text{saida}] = \text{oval}(\text{entrada}, \text{level}, \text{cor})$ - mascaramento oval

```
6
     % com o tamanho do triângulo definido por level e a cor do
 7
     \% mascaramento definido por cor.
8
9
     pos = 1;
10
     for blevel = level: -1:1,
11
12
        for t = 1: blevel,
          entrada(t, pos) = cor;
13
14
          entrada(t, 20 - pos) = cor;
15
          entrada(20-t, pos) = cor;
16
          \operatorname{entrada}(20 - t, 20 - \operatorname{pos}) = \operatorname{cor};
17
        end
18
        pos = pos + 1;
19
     end
20
21
     saida = entrada;
```

Rotina 12: pulapontosclust.m

```
function [perclust, saidapos] =
1
2
    pulapontosclust(entradax, entraday, precisao, numclust);
3
    % Executa a redução dos pontos de borda pele, assim como descrito
4
    \% na seção 6.2.1.
5
6
    % [perclust, saidapos] = pulapontosclust(entradax, entraday, precisao, numclust)
7
    \%entradax e entraday são os vetores x e y que representam os pontos.
8
    % precisao é a precisão da redução
9
    \%num<br/>clust é o número de clusters do algoritmo K-means.
10
11
     [centers, perclust] = kmeans_semfiguras([entradax, entraday],
12
        numclust);
13
    saidapos = [];
14
    percl = [];
15
16
17
    for t=1:numclust,
18
         pos = find(perclust == t);
19
20
         entradax_clust = entradax(pos);
21
         entraday_clust = entraday(pos);
22
23
         pts_clust = pulapontos(entradax_clust, entraday_clust,
             precisao);
```

```
24
25 saidapos = [saidapos, pos(pts_clust)];
26 percl = [percl,t*ones(size(pos(pts_clust)))];
27 end
```

```
Rotina 13: reduzirnetflipretang.m
```

```
1
    function [saida] = reduzirnetflipretang(rect, net, fig)
\mathbf{2}
    % Executa a redução por espelhamento, como descrito
3
    % na seção 6.4.1.1.
4
5
    \% [saida] = reduzirnetflipretang(rect, net, fig)
6
7
    % saida é a saída e rect é a entrada do ajuste fino, ambas
    \% correspondem a um vetor n \times 4 com todas as n detecções
8
    \% de faces no formato [x y altura largura].
9
    \% net é a rede neural e fig é a imagem original em tons de cinza.
10
11
12
    modo = net.userdata.modo;
13
14
    ystart = rect(:,1);
15
    yend = rect(:,3) + rect(:,1);
     xstart = rect(:,2);
16
17
    xend = rect(:,2) + rect(:,4);
18
     figtest = splitimage2facedetectionflip(fig,modo,xstart',xend
19
        ', ystart', yend');
20
21
     result = sim(net, figtest);
22
    pos = find((result(1,:) > 0.5) \& result(2,:) < result(1,:));
23
24
25
    saida = rect (pos, :);
```

Rotina 14: reduzirnetretang.m

```
1 function [saida] = reduzirnetretang(rect, net, fig,d)
2
```

- 3 % Executa o ajuste fino redução por recomparação sucessiva,
- $4 \quad \% \ {\rm como} \ {\rm descrito} \ {\rm na} \ {\rm se}{\rm cao} \ 6.4.1.3.$
- 5
- $6 \quad \% \text{ [saida]} = \text{reduzirnetretang}(\text{rect,net,fig})$
- 7 %saida é a saída e rect é a entrada do ajuste fino, ambas

```
8
    \% correspondem a um vetor n \times 4 com todas as n detecções
9
    % de faces no formato [x y altura largura].
    \% net é a rede neural e fig é a imagem original em tons de cinza.
10
    % d é o nível da recomparação sucessiva, ou seja, quantos pixels
11
12
    % haverá de liberdade para executar o ajuste.
13
14
    saida = rect;
    x = size(rect, 1);
15
16
    continua = 1;
17
    modo = net.userdata.modo;
18
    saida(1,5) = 0;
19
20
    for t = 1:x,
21
         maximo_local = saida(t, 5);
22
         while (\text{continua} = 1),
              ystart = saida(t, 1);
23
24
              yend = saida(t,3) + saida(t,1);
25
              x start = saida(t, 2);
26
              xend = saida(t,2) + saida(t,4);
              tamx = size(fig, 1);
27
28
              tamy = size(fig, 2);
29
30
              [xstart, xend, ystart, yend] =
31
    variacoesmaismenos(xstart, xend, ystart, yend, tamx, tamy, d);
32
33
              figtest = splitimage2facedetection (fig, modo, xstart,
                  xend, ystart, yend);
34
35
              result = comite(net, figtest);
36
37
              [\max \max, \operatorname{pos}] = \max(\operatorname{result}(1, :));
38
39
              if (maximo > maximo_local),
40
                   saida(t,:) =
     [ystart(pos), xstart(pos), yend(pos)-ystart(pos), xend(pos)-
41
        xstart(pos)
42
     , maximo];
43
                   maximo_local = maximo;
44
              else
45
                   d = floor(d/2);
                   if d == 0,
46
47
                       continua = 0;
48
                   end;
49
              end;
```

 $\begin{array}{ll} 50 & \text{end} \\ 51 & \text{continua} = 1; \\ 52 & \text{end} \end{array}$

Rotina 15: rgb2lab.m

```
1
    function [L, a, b] = rgb2lab(R, G, B)
2
3
    \% function [L,a,b] = rgb2lab(R,G,B)
4
    % Transforma o espaço de cores de RGB para LAB,
5
6
    \% como mostrado na seção 5.4.1.4.
7
8
    % rgb2lab recebe matrizes que representam R, G, and B
9
    % e os transforma no espaço CIE LAB. Originalmente escrito por Mark Ruzon.
10
11
    \% [L,a,b] = rgb2lab(R,G,B)
12
    % L,a e b são os valores de saída em LAB. R,G e B são os valores RGB.
13
    B = double(R(:,:,3))/255;
14
15
    G = double(R(:,:,2))/255;
16
    R = double(R(:,:,1))/255;
17
18
    [M, N] = size(R);
19
    s = M*N;
20
21
    % Limiar (recomendação ITU BT.709)
22
    T = 0.008856;
23
24
    RGB = [reshape(R,1,s); reshape(G,1,s); reshape(B,1,s)];
25
    \% Converte para XYZ
26
27
    MAT = [0.43395275530954]
                                   0.37621941468078
                                                         0.18982783001004
28
        0.2126710000026
                                                    0.0721690000003
                              0.71515999999967
29
        0.01775791409257
                              0.10947652086686
                                                     0.87276556504053];
30
    XYZ = MAT * RGB;
31
32
33
    XYZ3 = fastrootcube(XYZ);
34
35
    XYZT = XYZ > T;
36
37
    XT = XYZT(1, :);
38
    YT = XYZT(2,:);
```

```
39
    ZT = XYZT(3, :);
40
41
    X3 = XYZ3(1, :);
    Y3 = XYZ3(2,:);
42
     Z3 = XYZ3(3, :);
43
44
45
     fX = XT \cdot X3 + (~XT) \cdot (7.787 \times XYZ(1, :) + 16/116);
46
     fY = YT \cdot * Y3 + (~YT) \cdot * (7.787 * XYZ(2,:) + 16/116);
47
48
     L = YT .* (116*Y3 - 16.0) + (~YT) .* (903.3*XYZ(2,:));
49
     fZ = ZT \cdot Z3 + (~ZT) \cdot (7.787 * XYZ(3, :) + 16/116);
50
51
52
     a = 500 * (fX - fY);
53
     b = 200 * (fY - fZ);
54
55
    L = reshape(L, M, N);
     a = reshape(a, M, N);
56
57
     b = reshape(b, M, N);
58
59
    \mathbf{L} = \operatorname{cat}(3, \mathbf{L}, \mathbf{a}, \mathbf{b});
```

Rotina 16: rgblabel.m

1	function saida = rgblabel(filename, limiar, opt);
2	
3	% Executa uma segmentação por componentes conexos
4	%em uma figura colorida no arquivo filename.
5	
6	% O algoritmo de componentes con exos é aplicado sobre
7	%uma imagem binarizada a partir de um ponto que o usuário
8	% seleciona com o mouse durante o script.
9	
10	%Este script é utilizado para compor um banco de dados
11	% de pontos de pele e não-pele, como descrito na seção
12	% 6.1.1.1.
13	
14	% saida = rgblabel(filename,limiar,opt);
15	%saida é a imagem extraida do processo.
16	%Filename é o nome do arquivo da imagem
17	%Limiar é o valor utilizado para gerar a
18	% faixa +limiar -limiar com centro no valor
19	% RGB selecionado pelo usuário.

20 % opt é uma opção em cortar a imagem final ao retângulo

```
21
    % mínimo ou manter a imagem no tamanho original (opt='nocrop').
22
    if exist ('opt') == 0,
23
24
         opt = '';
25
    end
26
27
    if exist ('limiar') == 0,
28
         limiar = double(30);
29
    end
30
31
    if strcmp(class(filename), 'char'),
32
         entrada = imread(filename);
33
    else entrada = filename; end;
34
35
    imshow(entrada);
36
37
    \% Usuário seleciona 1 ponto
38
    [py, px, button] = ginput(1);
39
    if (button = 1),
40
41
        py = round(py);
42
         px = round(px);
43
44
         rponto = double(entrada(px, py, 1));
45
         gponto = double(entrada(px, py, 2));
46
         bponto = double(entrada(px, py, 3));
47
48
        R = entrada(:,:,1);
49
        G = entrada(:,:,2);
50
        B = entrada(:,:,3);
51
52
        % Binariza a imagem de acordo com limiar e o ponto selecionado
         Rlim = ((R < (rponto + limiar)) \& (R > (rponto - limiar))
53
            );
         Glim = ((G < (gponto + limiar)) \& (G > (gponto - limiar))
54
            );
         Blim = ((B < (bponto + limiar)) \& (B > (bponto - limiar))
55
            );
56
57
         fig = Rlim \& Glim \& Blim;
58
59
         pos = bwlabel(fig, 4);
60
61
         labelponto = pos(px, py);
```

```
62
63
         objeto = (pos == labelponto);
64
65
        R = double(R) .* double(objeto);
66
        G = double(G) .* double(objeto);
         B = double(B) .* double(objeto);
67
68
69
         saida (:,:,1) = R;
70
         saida (:,:,2) = G;
         saida (:,:,3) = B;
71
72
73
        % recortando apenas a seleção do usuário
         if strcmp(opt, 'nocrop') = 0,
74
             [posy, posx] = find(objeto ~= 0);
75
76
             \max = \max(posx);
77
             \min x = \min(posx);
78
             \max = \max(posy);
79
             \min y = \min(posy);
80
             saida = imcrop(saida, [minx, miny, maxx-minx, maxy-miny])
                 ;
81
         end
82
83
         saida = uint8(saida);
84
    else
         saida = [];
85
86
         if button == 100,
87
             delete (filename);
88
         end
89
    end
```

Rotina 17: reshapeimg2net.m

```
1
    function saida = reshapeimg2net(entrada);
\mathbf{2}
3
    % Função que recebe uma imagem e a padroniza para ser aceita como
    \%entrada de uma rede neural
4
5
6
    \% saida = reshapeimg2net(entrada);
 7
    % saida é a imagem padronizada para ser alimentada
    % a uma rede neural especializada em detecção de pele.
8
    % entrada é a matriz de imagem (RGB).
9
10
    xy = size(entrada, 1) * size(entrada, 2);
11
12
```

```
 \begin{array}{ll} 13 & \operatorname{entradaR} = \operatorname{entrada}\left(:\,,:\,,1\right); \\ 14 & \operatorname{entradaG} = \operatorname{entrada}\left(:\,,:\,,2\right); \\ 15 & \operatorname{entradaB} = \operatorname{entrada}\left(:\,,:\,,3\right); \\ 16 \\ 17 & \operatorname{saida} = \left[\operatorname{reshape}\left(\operatorname{entradaR}\,,1\,,\operatorname{xy}\right); \operatorname{reshape}\left(\operatorname{entradaG}\,,1\,,\operatorname{xy}\right); \\ & \operatorname{reshape}\left(\operatorname{entradaB}\,,1\,,\operatorname{xy}\right)\right]; \\ 18 & \operatorname{saida} = \operatorname{double}\left(\operatorname{saida}\right)/255; \end{array}
```

Rotina 18: tiraareas.m

```
1
    function saida = tiraareas(entrada, limiar);
2
3
    % Função que implementa a remoção de pequenas áreas
4
    \% saida = tiraareas(entrada, limiar);
5
    % entrada é a imagem contendo o filtro de pele com pele em branco
6
    % e não-pele em preto. Limiar é a área mínima permitida.
7
8
    pos = bwlabel(entrada, 8);
9
10
11
    tamhist = max(max(pos));
12
13
    if tamhist > 1,
14
         numcount = imhistc(pos, tamhist+1, 1, tamhist)';
    else numcount = [0; size(find(pos == 1), 1)]; end;
15
16
17
    numcount = numcount (2: end);
18
19
    areas = find (numcount < limiar);
20
21
    for t=1:size(areas,2),
        entrada(pos == areas(t)) = 0;
22
23
    end
24
25
    saida = entrada;
```

Rotina 19: variacoesmaismenos.m

```
1 function [xstartarr, xendarr, ystartarr, yendarr] =
2 variacoesmaismenos(xstart, xend, ystart, yend, tamx, tamy, d)
3
4 % Função que recebe 4 pontos e retorna uma lista de pontos
5 % contendo todos os pontos dentro de um seleção com tamanho d
```

```
6
7
    % Função utilizada em conjunto com a recomparação sucessiva.
8
9
    % function [xstartarr, xendarr, ystartarr, yendarr] =
10
   % variacoesmaismenos(xstart, xend, ystart, yend, tamx, tamy, d)
    % [xstartarr,xendarr,ystartarr,yendarr] são os vetores que contém todos os
11
12
    \% pontos pontos x inicial e final, y inicial e final.
    % [xstart,xend,ystart,yend] é o ponto inicial e d é o
13
    % tamanho do grau de liberdade da recomparação sucessiva.
14
15
16
    n = 1;
17
18
     for z = -1:1,
19
         for w = -1:1,
20
               for r = -1:1,
21
                    for s = -1:1,
22
                      x startarr(n) = x start + (z*d);
23
                      xendarr(n) = xend+(w*d);
24
                      ystartarr(n) = ystart+(r*d);
25
                      yendarr(n) = yend+(s*d);
26
                      n = n + 1;
27
                   end
28
              end
29
         end
30
    end
31
32
    pos1 = find(xstartarr <= 0);
33
    pos2 = find(xstartarr > tamx);
    pos3 = find(ystartarr <= 0);
34
35
    pos4 = find(ystartarr > tamy);
36
    pos5 = find(xendarr <= 0);
37
    pos6 = find(xendarr > tamx);
38
    pos7 = find(yendarr <= 0);
39
    pos8 = find(yendarr > tamy);
40
41
    pos = [pos1, pos2, pos3, pos4, pos5, pos6, pos7, pos8];
42
    x startarr(pos) = [];
    ystartarr(pos) = [];
43
44
    \operatorname{xendarr}(\operatorname{pos}) = [];
    yendarr(pos) = [];
45
```

Referências Bibliográficas

ADLER, F., Physiology of the Eye, Mosby Publishing, 1965

AHLBERG, J., A System for Face Localization and Facial Feature Extraction, Linkoping University, relatório técnico, 1999

ALBIOL, A., TORRES, L. e DELP, E., **Optimum Color Spaces for Skin Detection**, Proceedings of the International Conference on Image Processing p. 122–124, 2001

ANAGNOSTOPOULOS, I., ANAGNOSTOPOULOS, C., VERGADOS, D., LOUMOS, V. e KAYAFAS, E., A Probabilistic Neural Network for Human Face Identification based on Fuzzy Logic Chromatic Rules, IEEE MED03, 2003

ARCA, S., CAMPADELLI, P. e LANZAROTTI, R., An Automatic Feature-Based Face Recognition System, Dipartimento di Scienze dell'Informazione, relatório técnico, 2003

BEEK, P. V., REINDERS, M., SANKUR, B. e LUBBE, J. V. D., Semantic segmentation of videophone image sequences, SPIE'92 Vol. 1818 Visual Communications and Image Processing Boston, JUSA p. 1182–1193, 1992

BELHUMEUR, P. N., HESPANHA, J. P. e KRIEGMAN, D. J., Eigenfaces vs.
Fisherfaces: Recognition Using Class Specific Linear Projection, IEEE
Trans. Pattern Analysis and Machine Intelligence, volume 19, número 7, 1997

BERTILLON, A., La couleur de l'iris, Rev. Sci., vol. 36, no. 3, p. 65–73, 1885

BRAND, J. e MASON, J., A Skin Probability Map and its Use in Face **Detection**, Proceedings of the 2001 IEEE International Conference On Image Processing (ICIP-01) p. 1034–1037, 2001

BROWN, D., CRAW, I. e LEWTHWAITE, J., A SOM Based Approach to Skin **Detection with Application in Real Time Systems**, Mathematical Sciences, University of Aberdeen, relatório técnico, 2001

BURRUS, C. S., GOPINATH, R. A. e GUO, H., Introduction to Wavelets and Wavelet Transforms: A Primer, Prentice Hall, 1998

CANNY, J., A computational approach to edge detection, IEEE Trans. Pattern Anal. Machine Intelligence, volume 8, p. 679–698, 1986

CHEN, H., BHANU, B. e WANG, R., **Performance Evaluation and Prediction** for **3D Ear Recognition**, AVBPA Audio and Video-based Biometric Person Authentication 2005, p. 748–757, 2005

CHEN, L., ZHOU, J., LIU, Z., CHEN, W. e XIONG, G., A skin detector based on neural network, Sichuan University, China, relatório técnico, 2002

COLLINS, R. T., GROSS, R. e SHI, J., Silhouette-based Human Identification from Body Shape and Gait, Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition 2002 (FGR 2002), 2002

COTTON, S. D., **Do all human skin colours lie on a defined surface within LMS space?**, University of Birmingham, School of Computer Science, relatório técnico CSR-96-01, 1996

CRAW, I., ELIS, H. e LISHMAN, J. R., Automatic extraction of face-features, Pattern Recognition Letters, volume 5, número 2, p. 183–187, ISSN: 0167-8655, 1987

CRAW, I., TOCK, D. e BENNETT, A., **Finding Face Features**, European Conference on Computer Vision, p. 92–96, 1992

CROWLEY, J. e BERARD, F., Multi-Modal Tracking of Faces for Video Communications, Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), p. 640, IEEE Computer Society, ISBN: 0-8186-7822-4, 1997

DANNER, T. e DATTA, I., **ELEC 301 project: Eigenfaces Group**, Institute of computer Science of Rice University, relatório técnico, 1999

DAUGMAN, J. G., **High Confidence Visual Recognition of Persons by a Test of Statistical Independence**, IEEE Trans. Pattern Anal. Mach. Intell., volume 15, número 11, p. 1148–1161, 1993

DAVIS, L., Handbook of Genetic Algorithms, Van Nostrand Reingold, 1991

DE SILVA, L., AIZAWA, K. e HATORI, M., **Detection and Tracking of Facial Features by using a Facial Feature Model and Deformable Circular Templates**, IEICE Transactions on Information and Systems, Vol.E78-D, No.9, p. 1195–1207, IEEE Computer Society, 1995

DEL SOLAR, J. R. e VERSCHAE, R., **Skin Detection using Neighborhood Information**, Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004

DEMPSTER, A., A generalization of Bayesian inference, Journal of the Royal Statistical Society Series B 30 p. 205–247, 1968

DEMUTH, H. e BEALE, M., Neural Network Toolbox: For use with MATLAB: User's Guide, Mathworks, 2001

EDSON J. R. JUSTINO, F. B. e SABOURIN, R., Os Sistemas de Reconhecimento de Assinaturas sob a Ótica da Grafologia, ISDM'98, International Symposium of Document Manager - Curitiba - Brazil p. 242–254, 1998

ELLEN, D., The Scientific Examination of Documents: Methods and Techniques, Callaghan and Company, 1956

FAHLMAN, S. E., An Empirical Study of Learning Speed in Back-Propagation Networks, Carnegie Mellon University, relatório técnico CMU-CS-88-162, 1988

FARKAS, L. e MUNRO, I., Anthropometric Facial Proportions in Medicine, Charles C. Thomas, Publisher Ltd, 1987

FERET, **The FERET (Facial Recognition Technology) Database**, 2005. Disponível em www.itl.nist.gov/iad/humanid/feret /, acesso em 10 out. 2005 FERIS, R. S., DE CAMPOS, T. E. e JUNIOR, R. M. C., **Detection and Tracking** of Facial Features in Video Sequences, Lecture Notes in Artificial Intelligence, vol. 1793, 2000

FLOM, L. e SAFIR, A., Iris Recognition system, U.S. Patent 4 641 394, 1987

FRANZINI, M. A., WITBROCK, M. J. e LEE, K.-F., **Speaker-Independent Recognition of Connected Utterances Using Recurrent and Nonrecurrent Neural Networks**, IEEE International Joint Conference on Neural Networks (3rd IJCNN'89), IEEE, 1989

GALTON, F., **Personal Identification and description**, Nature magazine, June 21, p. 173–177, 1888

GARCIA, C., SIM, G. e TZIRITAS, G., A Feature-Based Face Detector Using Wavelet Frames, Department of Computer Science, University of Crete, relatório técnico, 2001

GEMERT, M. J. C. V., JACQUES, S. L., STERENBORG, H. J. C. M. e STAR, W. M., **Skin optics**, IEEE Trans. on Biomedical Engineering, ISBN: 0-599-52020-5, 1999

GERTZE, D., Automatic speaker verification using dynamic time warping and artificial neural networks, dissertação de mestrado, Clarkson University, 1999

GNANADESIKAN, R., Methods for Statistical Data Analysis of Multivariate Observations, John Wiley, New York, 1977

GONZALEZ, R. C. e WOODS, R. E., **Digital Image Processing**, Addison-Wesley Longman Publishing Co., Inc., 1992

GOOGLE, **Google, Inc.**, 2400 Bayshore Pkwy Mountain View, CA 94043 USA, 2005. Disponível em http://www.google.com, acesso em 10 out. 2005

GOVINDARAJU, V., Locating human faces in photographs, Int. J. Comput. Vision, volume 19, número 2, p. 129–146, ISSN: 0920-5691, 1996

GRAF, H. P., CHEN, T., PETAJAN, E. e COSATTO, E., Locating Faces and Facial Parts, AT&T Bell Laboratories, relatório técnico, 1994

GRAF, H. P., COSATTO, E., GIBBON, D., KOCHEISEN, M. e PETAJA, E., **Multi-Modal System for Locating Heads and Faces**, Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96), p. 88, IEEE Computer Society, ISBN: 0-8186-7713-9, 1996

GROB, B., Televisão Básica, Ed. Guanabara Dois, 1989

HANNA, K., MANDELBAUM, R., WIXSON, L., MISHRA, D. e PARAGANA, V., **A system for nonintrusive human iris acquisition**, Proceedings of The IEEE, Vol. 85, No. 9, September 1997, p. 200–203, 1996

HARRISON, W. R., Suspect Documents, Praeger, N.Y., 1958

HARTMAN, M., **Compact fingerprint scanner techniques**, Proceedings of Biometric Consortium 8th Meeting, 1996

HASTIE, T., KISHON, E., CLARK, M. e FAN, J., A model for signature verification, AT&T Bell Laboratories, relatório técnico, 1992

HAYKIN, S., Neural Networks: A Comprehensive Foundation, Prentice Hall PTR, 1994

HENRY, E., **Classification and Uses of Finger Prints**, Routledge, London, relatório técnico, 1900

HERPERS, R., KATTNER, H., RODAX, H. e SOMMER, G., GAZE: An attentive processing strategy to detect and analyze the prominent facial regions, International workshop on face and gesture recognition, 1995

HIGGINS, J. E., DODD, T. J. e DAMPER, R. I., Application of Multiple Classifier Techniques to Subband Speaker Identification with an HMM / ANN System, Lecture Notes in Computer Science, volume 2096, ISSN: 0302-9743, 2001

HSUAN YANG, M. e AHUJA, N., Gaussian Mixture Model for Human Skin Color and Its Applications in Image and Video Databases, The Pennsylvania State University CiteSeer Archives, 1998

HUANG, J., GUTTA, S. e WECHSLER, H., **Detection of human faces using decision trees**, Proceedings of Second International Conference on Automatic Face and Gesture Recognition. IEEE Computer Society Press, October 1996., p. 248–252, 1996

HUMANSCAN, **BioID** (Biological Identification) Face Database, 2005. Disponível em http://www.humanscan.de/, acesso em 10 out. 2005

IP, H. H.-S. e NG, J. M. C., Human Face Recognition using Dempster-Shafer Theory., Proceedings of International Conference on Image Processing, 1994

JAIN, A. K., Fundamentals of digital image processing, Prentice-Hall, Inc., 1989

JAIN, A. K., CHEN, H. e MINUT, S., **Dental Biometrics: Human Identification Using Dental Radiographs**, AVBPA Audio and Video-based Biometric Person Authentication 2003, p. 429–437, 2003

JAIN, A. K. e DUTA, N., **Deformable Matching of Hand Shapes for User Verification**, ICIP The IEEE International Conference on Image Processing, p. 857–861, 1999

JAIN, A. K., PANKANTI, S. e ROSS, A., **A Prototype Hand Geometry-based Verification System**, Proc. of 2nd International Conference on Audio and Videobased Biometric Person Authentication (AVBPA), Washington D.C., p. 166–171, 1999

JANG, J. S. R., SUN, C. T. e MIZUTANI, E., Neuro-Fuzzy and Soft Computing, Prentice Hall, 1997

JESORSKY, O., KIRCHBERG, K. J. e FRISCHHOLZ, R. W., Robust Face **Detection Using the Hausdorff Distance**, Lecture Notes in Computer Science, volume 2091, ISSN: 0302-9743, 2001

JOHNSON, R. A. e WICHERN, D. W., Applied multivariate statistical analysis, Prentice-Hall, Inc., 1988

JOHNSON, R. G., **Can iris patterns be used to identify people, Los Alamos National Laboratory**, Los Alamos National Laboratory, CA, Chemical and Laser Sciences Division, relatório técnico, 1991

JONES, M. J. e REHG, J. M., Statistical Color Models with Applications to Skin Detection, Hewlett Packard Laboratories, relatório técnico CRL-98-11, 1998

KANADE, T., Picture Processing System by Computer Complex and Recognition of Human Faces, doctoral dissertation, Kyoto University, 1973

KJELDSEN, R. e KENDER, J., Finding skin in color images, Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96), p. 312, IEEE Computer Society, ISBN: 0-8186-7713-9, 1996

KRONFELD, P. C., **The gross anatomy and embryology of the eye**, The Eye, vol. 1, p. 1–66, Ed. London Academic, 1968

KRUPPAAND, H., BAUER, M. e SCHIELE, B., Skin patch detection in realworld images, Perceptual Computing and Computer Vision Group, ETH Zurich, Switzerland, relatório técnico, 2002

KUMAR, A., WONG, D. C. M., SHEN, H. C. e JAIN, A. K., **Personal** Verification Using Palmprint and Hand Geometry Biometric, AVBPA Audio and Video-based Biometric Person Authentication 2003, p. 668–678, 2003

LADES, H. M., BARTLETT, M. S. e SEJNOWSKI, T. J., **Independent Component Representations for Face Recognition**, Proceedings of SPIE, 1998, vol. 3299, pp. 528-539., 1998

LAZLI, L. e SELLAMI, M., **Speaker independent isolated speech recognition for Arabic language using hybrid HMM-MLP-FCM system**, Book of Abstracts of International Conference on Computer Systems and Applications p. 108, 2003 LEE, C.-S. e ELGAMMAL, A. M., Gait Style and Gait Content: Bilinear Models for Gait Recognition Using Gait Re-sampling, Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition, p. 147–152, 2004

LEE, H. C. e GAENSSLEN, R. E., Advances in Fingerprint Technology, Elsevier, New York, 1991

LEGGETT, J. e WILLIAMS, G., Verifying Identity via Keystroke Characteristics, International Journal of Man-Machine Studies, volume 28, número 1, p. 67–76, 1988

LEUNG, W. F., LAU, W. H., LEUNG, S. H. e LUK, A., Fingerprint Recognition Using Neural Network, Proc. of the IEEE Workshop Neural Networks for Signal Processing, p. 226–235, 1991

LI, W.-Y. e O'SHAUGHNESSY, D., **Hybrid Networks Based on RBFN and GMM for Speaker Recognition**, Proc. Eurospeech '97, p. 955–958, Rhodes, Greece, 1997

LIAO, R. e LI, S. Z., Face Recognition Based on Multiple Facial Features, 4th IEEE International Conference on Automatic Face and Gesture Recognition, 2000

LIU, C. e WECHSLER, H., Gabor Feature Based Classification Using the Enhanced Fisher Linear Discriminant Model for Face Recognition, IEEE Trans. Image Processing, vol. 11, no. 4, pp. 467476, p. 467–476, 2002

LUENBERGER, D. G., Introduction to Linear and Nonlinear Programming, Addison-Wesley, New York, NY, USA, 1973

MALTONI, D., MAIO, D., JAIN, A. K. e PRABHAKAR, S., Handbook of Fingerprint Recognition, 1 ed., Springer Verlag New York, 1991

MARQUES, A. C. P. B., Extração de Minúcias em Imagens de Impressões Digitais utilizando Redes Neurais, dissertação de mestrado, Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro, Orientador: Antonio Carlos Gay Thomé, 2004

MATHWORKS, T., **Matlab**, 2005. Disponível em http://www.mathworks.com, acesso em 10 out. 2005

MCCULLOCH, W. S. e PITTS, W. H., **A logical calculus of the ideas immanent in nervous activity**, Bulletin of Mathematical Biophysics, p. 115– 133, 1943

MCKENNA, S., GONG, S. e RAJA, Y., Modelling facial colour and identity with gaussian mixtures, Departament of Applied Computing, University of Dundee, Scotland, relatório técnico, 1996

MOGHADDAM, B. e PENTLAND, A., **Probabilistic Visual Learning for Object Representation**, IEEE Trans. Pattern Anal. Mach. Intell., volume 19, número 7, p. 696–710, ISSN: 0162-8828, 1997

MONTGOMERY, D. C., PECK, E. A. e VINING, G. G., Introduction to Linear Regression Analysis, Jossey-bass, 2001

MORRIS, ZAIDI, B., **Blink detection for real-time eye tracking**, Journal of Networking and Computer Applications, volume 25, p. 129–143, 2002

MOUTINHO, A. M. e THOMÉ, A. C. G., Localizador de Padrões Faciais: Uma Aplicação usando-se Redes Neurais Artificiais, Engevista, Revista da Engenharia da UFF, 2003

NAKAGAWA, S., UEDA, Y. e SEINO, T., **Speaker-independent, Text-independent Language Identification by HMM**, Proc. ICSLP '92, Banff, Canada, 1992

NETO, A. L., **Extração automática de minúcias de impressões digitais**, dissertação de mestrado, UFF - Universidade Federal Fluminense, 2000

PANKANTI, S., PRABHAKAR, S. e JAIN, A. K., Fingerprint Recognition Using Neural Network Vol. 24, No. 8, IEEE Transactions on PAMI, p. 1010– 1025, 2002 PARIZEAU, M. e PLAMONDON, R., A Comparative Analysis of Regional Correlation, Dynamic Time Warping, and Skeletal Tree Matching for Signature Verification, IEEE Trans. Pattern Anal. Mach. Intell., volume 12, número 7, p. 710–717, ISSN: 0162-8828, 1990

PEER, P., KOVAC, J. e SOLINA, F., **Human Skin Colour Clustering for Face Detection**, EUROCON 2003 - International Conference on Computer as a Tool, 2003

PENTLAND, A., MOGHADDAM, B. e STARNER, T., View-based and modular eigenspaces for face recognition, IEEE Conference on Computer Vision and Pattern Recognition, 1994

PRASAD, A. e AMARESH, V., An Offline Signature Verification System, University of Arizona - Eletrical and Computer Engineering, relatório técnico, 2004

PRATT, W. K., **Digital Image Processing (3nd Edition)**, John Wiley & Sons, Inc., New York, New York, 2001

PRTEUX, F. J., GENEST, G. e SABOURIN, R., Off-Line Signature Verification by Local Granulometric Size Distributions, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 9, September., ISSN: 00615447, 1997

RABINER, L. R., A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, volume 77, número 2, p. 257–286, 1989

ROSE, R. e REYNOLDS, D., **Text independent speaker identification using automatic acoustic segmentation**, Proceedings of ICASSP-90 International Conference on Acoustics, Speech, and Signal Processing, Vol. 1, p. 293–296, IEEE, 1990

ROSENBLATT, F., The perceptron: a probabilistic model for information storage and organization in the brain, Psychological Review, p. 386–408, 1958 ROWLEY, H., BALUJA, S. e KANADE, T., Human Face Detection in Visual Scenes, Advances in Neural Information Processing Systems Number 8, 1996

ROWLEY, H., BALUJA, S. e KANADE, T., Rotation Invariant Neural Network-Based Face Detection, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p. 38, IEEE Computer Society, ISBN: 0-8186-8497-6, 1998a

ROWLEY, H. A., BALUJA, S. e KANADE, T., Neural Network-Based Face Detection, IEEE Trans. Pattern Anal. Mach. Intell., volume 20, número 1, p. 23–38, ISSN: 0162-8828, 1998b

ROWLEY, H. A. e KANADE, T., Neural network-based face detection, Tese de Ph.D., School of computer science, Carniege Mellon University, 1999

RUDASI, L., Text-independent automatic speaker identification using partitioned neural networks, Tese de Ph.D., Old Dominion University, 1992

RUMELHART, D. E., DURBIN, R., GOLDEN, R. e CHAUVIN, Y., Backpropagation: The Basic Theory, Y. Chauvin e D. E. Rumelhart (eds.), Backpropagation: Theory, architectures, and applications, Erlbaum, 1993

SAKAI, T., NAGAO, M. e KANADE, T., **Computer Analysis and Classification of Photographs of Human Faces**, Proc. First USA-JAPAN Computer Conference, p. 55–62, 1972

SANKOFF, D. e KRUSKAL, J., **Time warps, string edits, and macromolecules. Introduction by John Nerbonne**, The David Hume Series of Philosophy and Cognitive Science Reissues. Stanford, CA, 1999

SANTOS, C., JUSTINO, E. J. R., BORTOLOZZI, F. e SABOURIN, R., An Off-Line Signature Verification Method based on the Questioned Document Expert's Approach and a Neural Network Classifier, PUCPR - Pontifícia Universidade Católica do Paraná, relatório técnico, 2004 SCHMIDT, M., GOLDEN, J. e GISH, H., **GMM Sample Statistic Log-**Likelihoods for Text-Independent Speaker Recognition, Proc. Eurospeech '97, p. 855–858, Rhodes, Greece, 1997

SELLERS, P. H., **Pattern recognition in DNA.**, R. Miura (ed.), Some Mathematical Questions in Biology: DNA Sequence Analysis., Vol. 17 of Mathematics in the Life Sciences., p. 19–28, Am. Math. Soc., 1986

SENIOR, A., HSU, R.-L., MOTTALEB, M. A. e JAIN, A. K., Face Detection in Color Images, IEEE Trans. Pattern Anal. Mach. Intell. p. 696–706, ISSN: 0162-8828, 2002

SHAFER, G., A Mathematical Theory of Evidence, Princeton University Press, 1976

SHAHIN, I. e BOTROS, N., Speaker identification using dynamic time warping with stress compensation technique, Proceedings of Southeastcon '98, p. 65–68, IEEE, 1998

SIDNEY, U. N., Sistemas de Televisão e Vídeo, Editora LTC, 1991

SIROVICH, L. e KIRBY, M., Low-dimensional procedure for the characterization of human faces, Journal of the Optical Society of America, March, 1987

SPATH, H., Cluster dissection and analysis: theory, FORTRAN programs, examples, The Ellis Horwood series in computers and their applications ;, Halsted Press [distributor], New York, 1985

STAROVOITOV, V. e SAMAL, D., **A Geometric Approach to Face Recognition**, Institute of Engineering Cybernetics, Minsk, BELARUS, relatório técnico, 2000

TAKÁCS, B. e WECHSLER, H., Face Location Using A Dynamic Model of Retinal Feature Extraction, 1995 TAPPERT, C. C. e WAKAHARA, C. Y. S., **The state of the art in on-line handwriting recognition**, IEEE Transactions on Pattern Analysis and Machine Intelligence p. 787–807, 1990

TERRILLON, J.-C., FUKAMACHI, H., AKAMATSU, S. e SHIRAZI, M. N., Comparative Performance of Different Skin Chrominance Models and Chrominance Spaces for the Automatic Detection of Human Faces in Color Images, Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000, p. 54, IEEE Computer Society, ISBN: 0-7695-0580-5, 2000

THOMÉ, A. C. G., **Fundamentos sobre redes neurais artificiais parte 1**, Núcleo de computação eletrônica, Universidade federal do rio de Janeiro, relatório técnico, 2005

TOMAZ, F., CANDEIAS, T. e SHAHBAZKIA, H., **Improved Automatic Skin Detection in Color Images**, Proc. VIIth Digital Image Computing: Techniques and Applications, 2003

TURK, M. e PENTLAND, A., Face processing: models for recognition, SPIE Vol. 1192: Intelligent Robots and Computer Vision VIII: Algorithms and Techniques, 1989

TURK, M. e PENTLAND, A., **Eigenfaces for recognition**, Journal of Cognitive Neuroscience, 1991

VEZHNEVETS, V., SAZONOV, V. e ANDREEVA, A., **A Survey on Pixel-Based Skin Color Detection Techniques**, Faculty of Computational Mathematics and Cybernetics, 2003

VIANNA, G. K., RODRIGUES, R. J. e THOMÉ, A. C. G., Extração de Características para o Reconhecimento de Dígitos, Simpósio Brasileiro de redes neurais, 2000

WANG, L., CHEN, K. e CHI, H., Capture Inter-Speaker Information with a Neural Network for Speaker Identification, IJCNN '00: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)- Volume 5, p. 5247, IEEE Computer Society, Washington, DC, USA, ISBN: 0-7695-0619-4, 2000

WILDES, R., **Iris recognition: an emerging biometric technology**, IEEE Proceedings, vol. 85, p. 1–8, 1997

WILDES, R. P., ASMUTH, J. C., GREEN, G. L., HSU, S. C., KOLCZYNSKI, R. J., MATEY, J. R. e MCBRIDE, S. E., A machine vision system for iris recognition, Mach. Vision Applicat., vol. 9, p. 65–73, 1996

WISKOTT, L., FELLOUS, J.-M., KRÜGER, N. e VON DER MALSBURG, C., Face Recognition by Elastic Bunch Graph Matching, Institut für Neuroinformatik, relatório técnico, 1996

WISKOTT, L., FELLOUS, J.-M., KRÜGER, N. e VON DER MALSBURG, C., Face Recognition by Elastic Bunch Graph Matching, 7th Conference on Computer Analysis of Images and Patterns, CAIP'97, Kiel, p. 456–463, Springer-Verlag, 1997

WOHLBERG, B. e GREENE, J. R., Automatic on-line signature verification using Dynamic Time-Warping, Proceedings of the Third South African Workshop on Pattern Recognition, p. 185–192, Pretoria, South Africa, 1992

WUTIWIWATCHAI, C., ACHARIYAKULPORN, V. e TANPRASERT, C., Textdependent speaker identification using LPC and DTW for Thai language, TENCON 99. Proceedings of the IEEE Region 10 Conference. Volume 1, IEEE, 1999

YAHOO!, **Yahoo! do Brasil Internet Ltda.**, Rua Fidêncio Ramos 195 - 120. andar 04551-010 / São Paulo - SP, 2005. Disponível em http://www.yahoo.com, acesso em 10 out. 2005

YANG, J., LU, W. e WAIBEL, A., **Skin-Color Modeling and Adaptation**, Proceedings of the Third Asian Conference on Computer Vision-Volume II, p. 687– 694, Springer-Verlag, ISBN: 3-540-63931-4, 1997 YANG, J. e WAIBEL, A., **A real-time face tracker**, Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision (WACV '96), p. 142, IEEE Computer Society, ISBN: 0-8186-7620-5, 1996

YANG, M.-H., Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods, FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, p. 215, IEEE Computer Society, Washington, DC, USA, ISBN: 0-7695-1602-5, 2002

ZHAO, W., CHELLAPPA, R., PHILLIPS, P. e ROSENFELD, A., Face Recognition: A Literature Survey, Computer Based Learning Unit, University of Leeds, United kingdom, relatório técnico, 2004

ZHENG, H., DAOUDI, M. e JEDYNAK, B., **Blocking Adult Images Based on Statistical Skin Detection**, Electronic Letters on Computer Vision and Image Analysis, 2000

ZURADA, J. M., Introduction to Artificial Neural Systems, PWS Publishing Co., 1999