**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**Comparação entre métodos de normalização da iluminação
utilizados para melhorar a taxa do reconhecimento facial**

MICHELLE MAGALHÃES MENDONÇA

Dissertação apresentada à Escola de Engenharia de São Carlos, da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Orientadora: Prof^a. Dr^a. Maria Stela Veludo de Paiva.

São Carlos
2008

Ao meu pai, Luiz Alberto, e em memória a
minha mãe, Maria Angélica, por terem me
educado para eu me tornar a pessoa que sou
hoje.

Agradecimentos

Agradeço primeiramente a Deus por eu estar aqui realizando este trabalho.

Ao meu pai pela ajuda financeira e por acreditar em mim.

À minha família pelo incentivo para a realização do meu mestrado

À minha orientadora Dra. Maria Stela Veludo de Paiva pela confiança em mim, pela paciência e dedicação para me orientar.

À banca examinadora pelas dicas e correções desse trabalho.

À Universidade de São Paulo por colocar à minha disposição computadores e todo material necessário para a realização deste trabalho.

Aos meus amigos do Laboratório de Visão Computacional (LAVI), Gustavo, Juliana, Nilvana, Antônio e Ricardo Salvino, pela ajuda na correção de artigos, por tirar dúvidas minhas em relação a esta dissertação e pelo companheirismo durante período em que estava cursando as disciplinas. Também quero agradecer a Ricardo Fernandes pelo algoritmo de reconhecimento de faces que utiliza PCA.

À meu namorado Arnaldo Cândido Júnior pelo companheirismo e apoio.

Sou grata aos funcionários do Departamento de Engenharia Elétrica da EESC - USP, e aos professores da pós-graduação.

Resumo

MICHELLE M.M. (2008) Comparação entre métodos de normalização da Iluminação utilizados para melhorar a taxa do reconhecimento facial. Dissertação (Mestrado) Escola de Engenharia de São Carlos Universidade de São Carlos, 2008.

Condições distintas de iluminação numa imagem podem produzir representações desiguais do mesmo objeto, dificultando o processo de segmentação e reconhecimento de padrões, incluindo o reconhecimento facial. Devido a isso, a distribuição de iluminação numa imagem é considerada de grande importância, e novos algoritmos de normalização utilizando técnicas mais recentes ainda vêm sendo pesquisados. O objetivo dessa pesquisa foi o de avaliar os seguintes algoritmos de normalização da iluminação encontrados na literatura, que obtiveram bons resultados no reconhecimento de faces: LogAbout, variação do filtro homomórfico e método baseado em *wavelets*. O objetivo foi o de identificar o método de normalização da iluminação que resulta na melhor taxa de reconhecimento facial. Os algoritmos de reconhecimento utilizados foram: auto-faces, PCA (*Principal Component Analyses*) com rede neural LVQ (*Learning Vector Quantization*) e *wavelets* com rede neural MLP (*Multilayer Perceptron*). Como entrada, foram utilizadas imagens do banco Yale, que foram divididas em três subconjuntos. Os resultados mostraram que o método de normalização da iluminação que utiliza *wavelet* e LogAbout foram os que apresentaram melhoria significativa no reconhecimento facial. Os resultados também evidenciaram que, de uma maneira geral, com a utilização dos métodos de normalização da iluminação, obtém-se uma melhor taxa do reconhecimento facial, exceto para o método de normalização variação do filtro homomórfico com os algoritmos de reconhecimento facial auto-faces e *wavelet* com rede neural MLP.

Palavras – chave: métodos de normalização da iluminação, reconhecimento facial, *wavelets*, PCA, auto-faces, redes neurais artificiais.

Abstract

MICHELLE M.M. (2008) Comparison between illumination normalization methods used to improve the rate of facial recognition. Dissertation - Escola de Engenharia de São Carlos Universidade de São Carlos, 2008.

Distinct lighting conditions in an image can produce unequal representations of the same object, compromising segmentation and pattern recognition processes, including facial recognition. Hence, the lighting distribution on an image is considered of great importance, and normalization algorithms using new techniques have still been researched. This research aims to evaluate the following illumination normalization algorithms found in literature: LogAbout, variation of homomorphic filter and wavelet based method. The main interest was to find out the illumination normalization method which improves the facial recognition rate. The algorithms used for face recognition were: eigenfaces, PCA (Principal Component Analysis) with LVQ neural network and wavelets with MLP (Multilayer Perceptron) neural network. Images from Yale Face Database B, divided into three subsets have been used. The results show that the wavelet and LogAbout technique provided the best facial recognition rate. Experiments showed that the illumination normalization methods, in general, improve the facial recognition rate, except for the variation of homomorphic filter technique with the algorithms: eigenfaces and PCA with LVQ.

Keywords: illumination normalization methods, face recognition, wavelets, PCA, eigenfaces, artificial neural networks.

Lista de Figuras

Figura 1: Imagens da mesma pessoa submetida a diferentes condições de iluminação.....	3
Figura 2: Processamento no domínio da frequência.....	6
Figura 3: Curva dos filtros passa alta: a) Ideal. b) Butterworth. c) Gaussiano.....	7
Figura 4: Filtro Butterworth passa-baixa.....	10
Figura 5: Passos filtro homomórfico.	10
Figura 6: Aplicação do filtro Butterworth passa-alta: a) Imagem original. b) DFT. c) Imagem resultante com $D_0=0.5$ e $n=10$. d) Imagem resultante com $D_0=0.5$ e $n=25$	11
Figura 7: Transformada de Fourier aplicada num sinal estacionário (a), e não-estacionário (b).	12
Figura 8: Exemplo de funções base <i>wavelets</i>	15
Figura 9: Árvore de Decomposição <i>Wavelet</i>	16
Figura 10: Banco de filtros.	17
Figura 11: Esquema de decomposição bi-dimensional	18
Figura 12: Decomposição de uma imagem. PA e PB indicam filtros passa-alta e passa-baixa, respectivamente. A componente LL é utilizada como dado de entrada para o próximo nível. Esta arquitetura de filtragem é denominada de banco de filtros.....	19
Figura 13: Matriz de coeficientes da Daub 4.....	21
Figura 14: Matriz inversa.	22
Figura 15: Modelo de um neurônio artificial.....	23
Figura 16: Funções de ativação comuns.....	23
Figura 17: Arquitetura <i>feedforward</i> : a) De camada única. b) De múltiplas camadas.	24
Figura 18: Aprendizado supervisionado de uma rede neural.	25
Figura 19: a) Rede LVQ. b) Diagrama de Voronoi.....	29
Figura 20: Transformações não-lineares.	32
Figura 21: Método LogAbout.....	34
Figura 22: Normalização com o método LogAbout: (1) Imagem original. (2) Imagem resultante da transformação logarítmica. (3) Imagem resultante.	34
Figura 23: Imagem dividida em quatro regiões para normalização da iluminação.....	36
Figura 24: Resultados do método Estratégia Baseada em Região.....	37
Figura 25: Passos para re-escala da imagem.	39
Figura 26: Imagem original com diferentes iluminações na coluna 1, HE na segunda coluna e, resultado do método de correção de iluminação de Zhu et al. (2003) na terceira coluna.	41
Figura 27: Estrutura de multi-resolução de decomposição <i>wavelet</i>	42
Figura 28: Normalização com o método <i>wavelet</i> : (a) Imagens originais; (b) Imagens após a equalização do histograma. (c) Imagens resultantes.	43
Figura 29: Análise <i>Wavelet</i>	44
Figura 30: Alguns resultados do algoritmo que utiliza RNA utilizado para a compensação da iluminação.	45
Figura 31: Variação do filtro homomórfico.	46
Figura 32: Realce de imagem: (a) Imagem original, (b) Utilizando HE, (c) Utilizando filtro homomórfico, (d) Utilizando filtro homomórfico e realce de contraste, (e) Utilizando filtro homomórfico das duas sub-imagens verticais separadamente, (f) Utilizando filtro homomórfico das duas sub-imagens horizontais separadamente, (g) Utilizando a variação do filtro homomórfico.	47

Figura 33: Resultados da aplicação do SSR utilizando três funções <i>surround</i> : inverso do quadrado, exponencial e gaussiana.....	49
Figura 34: Realce de imagem com o algoritmo retinex: a) Imagem original b) Utilizando o método SSR c) Utilizando o método MSR.....	50
Figura 35: Realce de imagem com o MSRCR: a) Imagem original. b) Imagem após equalização do histograma. c) Resultado.	51
Figura 36: Visão geral de métodos baseados em aparência local.....	54
Figura 37: Imagem original do banco Yale Face Database B em (a), e imagem do banco Yale Face Database B após recorte da região de interesse em (b).....	61
Figura 38: Imagens de face de uma pessoa de cada subconjunto. a) Subconjunto 1. b) Subconjunto 2. c) Subconjunto 3.....	62
Figura 39: Diagrama com os algoritmos de normalização de iluminação e reconhecimento facial utilizados no trabalho.....	63
Figura 40: Máscara utilizada.	64
Figura 41: Normalização com o método LogAbout: a) Imagens originais. b) Imagens após transformação logarítmica. c) Imagens finais com alargamento de contraste, ou seja, imagens resultantes.	64
Figura 42: Imagens originais na primeira linha, e imagens após serem normalizadas com o método LogAbout na segunda linha.....	65
Figura 43: Decomposição <i>wavelet</i> de uma imagem: a) Imagem original. b) Primeiro nível de decomposição. c) Segundo nível de decomposição.....	65
Figura 44: Resultado final da decomposição <i>wavelet</i>	66
Figura 45: Normalização com o método <i>wavelet</i> : a) Imagens originais. b) Resultados.	67
Figura 46: Imagens originais em (a) e (d), imagens após método <i>wavelet</i> com fator de escala igual a 3 em (b) e (e), e imagens após método <i>wavelet</i> com fator de escala igual a 90 em (c) e (f).	67
Figura 47: Imagens originais na primeira linha, e imagens após serem normalizadas com o método variação do filtro homomórfico na segunda linha.	68
Figura 48: Gráfico do método que utiliza PCA e LVQ.....	71
Figura 49: Rede MLP do algoritmo de reconhecimento facial que utiliza <i>wavelets</i>	74
Figura 50: Gráfico do método que utiliza <i>wavelet</i> e MLP.	75
Figura 51: Gráfico do método que utiliza auto-faces.	78

Lista de tabelas

Tabela 1: Algoritmo de treinamento da rede LVQ.....	30
Tabela 2: Algoritmo de reconhecimento utilizando PCA e rede neural LVQ.....	57
Tabela 3: Tempo de processamento dos métodos de normalização de iluminação utilizados.....	69
Tabela 4: Resultado do método que utiliza PCA e LVQ.....	72
Tabela 5: Resultado do método que utiliza <i>wavelets</i> e RNA no subconjunto 3.....	75
Tabela 6: Resultado do método que utiliza auto-faces.....	78
Tabela 7: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 1 original.....	89
Tabela 8: Tabela de confusão com os resultados do reconhecimento com o algoritmo PCA com LVQ aplicados sobre o subconjunto 1 original.....	89
Tabela 9: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelets</i>	90
Tabela 10: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelets</i>	90
Tabela 11: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.....	90
Tabela 12: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.....	90
Tabela 13: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.....	91
Tabela 14: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.....	91
Tabela 15: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 2 original.....	91
Tabela 16: Tabela de confusão com os resultados do reconhecimento com o algoritmo PCA com LVQ aplicados sobre o subconjunto 2 original.....	91
Tabela 17: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelet</i>	92
Tabela 18: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelet</i>	92
Tabela 19: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.....	92
Tabela 20: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.....	92

Tabela 21: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.	93
Tabela 22: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.	93
Tabela 23: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 3 original.	93
Tabela 24: Tabela de confusão com os resultados do reconhecimento com o algoritmo PCA com LVQ aplicados sobre o subconjunto 3 original.	93
Tabela 25: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelet</i>	94
Tabela 26: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelet</i>	94
Tabela 27: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.	94
Tabela 28: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.	94
Tabela 29: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.	95
Tabela 30: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.	95
Tabela 31: Taxa de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 1 original.	96
Tabela 32: Tabela de confusão com os resultados do reconhecimento com o algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 1 original.	96
Tabela 33: Taxa de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelet</i>	97
Tabela 34: Tabela de confusão com os resultados de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelet</i>	97
Tabela 35: Taxa de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.	97
Tabela 36: Tabela de confusão com os resultados de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.	97
Tabela 37: Taxa de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.	98
Tabela 38: Tabela de confusão com os resultados de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.	98
Tabela 39: Taxa de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 2 original.	98

Tabela 40: Tabela de confusão com os resultados do reconhecimento com o algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 2 original.	98
Tabela 41: Taxa de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelet</i>	99
Tabela 42: Tabela de confusão com os resultados de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelet</i>	99
Tabela 43: Taxa de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.	99
Tabela 44: Tabela de confusão com os resultados de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.	99
Tabela 45: Taxa de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.	100
Tabela 46: Tabela de confusão com os resultados de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método filtro homomórfico.	100
Tabela 47: Taxa de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 3 original.	100
Tabela 48: Tabela de confusão com os resultados do reconhecimento com o algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 3 original.	100
Tabela 49: Taxa de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelet</i>	101
Tabela 50: Tabela de confusão com os resultados de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelet</i>	101
Tabela 51: Taxa de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.	101
Tabela 52: Tabela de confusão com os resultados de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.	101
Tabela 53: Taxa de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.	102
Tabela 54: Tabela de confusão com os resultados de reconhecimento do algoritmo <i>wavelets</i> com MLP aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.	102
Tabela 55: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-face aplicados sobre o subconjunto 1 original.	103
Tabela 56: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelet</i>	104
Tabela 57: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.	104

Tabela 58: Tabela com taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.	105
Tabela 59: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-face aplicados sobre o subconjunto 2 original.	105
Tabela 60: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelet</i>	106
Tabela 61: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.	106
Tabela 62: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.	107
Tabela 63: Tabela com a taxa de acerto para cada pessoa do reconhecimento com o algoritmo auto-face aplicados sobre o subconjunto 3 original.	107
Tabela 64: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método <i>wavelet</i>	108
Tabela 65: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.	108
Tabela 66: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.	109

Lista de abreviaturas e siglas

CDF	<i>Cumulative Distribution Function</i> (em português: Função de Distribuição Acumulada)
CWT	<i>Continuous Wavelets Transform</i> (em português: Transformada <i>Wavelet</i> Contínua)
DFT	<i>Discret Fourier Transform</i> (em português: Transformada Discreta de Fourier)
DWT	<i>Discrete Wavelet Transform</i> (em português: Transformada Discreta de <i>Wavelet</i>)
FT	<i>Fourier Transform</i> (em português: Transformada de Fourier)
GIC	<i>Gamma Intensity Correction</i> (em português: Correção de Intensidade Gama)
HE	<i>Histogram Equalization</i> (em português: Equalização do Histograma)
IDWT	<i>Inverse Discrete Wavelet Transform</i> (em português: Transformada <i>Wavelet</i> Discreta Inversa)
LVQ	<i>Learning Vector Quantization</i> (em português: Aprendizagem por Quantização de Vetor)
MATLAB	<i>Matrix Laboratory</i>
MLP	<i>Multilayer Perceptron</i> (em português: Perceptron Multicamadas)
PCA	<i>Principal Component Analysis</i> (em português: Análise do Componente Principal)
RNAs	<i>Artificial Neural Network</i> (em português: Redes Neurais Artificiais)
STFT	<i>Short Time Fourier Transform</i> (em português: Transformada de Fourier Janelada)

Sumário

RESUMO.....	i
ABSTRACT	ii
LISTAS DE FIGURAS	iii
LISTA DE TABELAS	v
LISTA DE ABREVIATURAS E SIGLAS	ix
1 Introdução	1
1.1 Proposta	3
1.2 Objetivo	4
1.3 Estrutura do trabalho	4
2 Fundamentação teórica	5
2.1 Considerações iniciais	5
2.2 Filtro Homomórfico.....	5
2.2.1 Filtros no domínio da frequência - conceitos.....	5
2.2.2 Filtragem Homomórfica.....	7
2.3 <i>Wavelets</i>	11
2.3.1 Transformada Discreta de <i>Wavelet</i>	12
2.3.2 Bases <i>Wavelets</i>	14
2.3.3 Análise Multiresolução	15
2.3.4 Transformada <i>Wavelet</i> bi-dimensional	17
2.3.5 <i>Wavelets</i> de Daubechies.....	19
2.4 Redes Neurais Artificiais.....	22
2.4.1 Aprendizado Supervisionado	24
2.4.2 Redes Perceptron de Múltiplas Camadas (MLP).....	25
2.4.2.1 Treinamento	28
2.4.3 <i>Learning Vector Quantization</i> (LVQ)	28
2.5 Considerações finais	30
3 Métodos para correção de iluminação	31
3.1 Considerações iniciais	31
3.2 Realce de imagens no domínio do espaço	31

3.2.1	Transformações nos níveis de cinza	32
3.2.2	Equalização do Histograma (HE) e Equalização Adaptativa de Histograma (AHE)	33
3.3	Método LogAbout	34
3.4	Correção de Intensidade Gama (GIC), Estratégia Baseada em Região e <i>Quotient Illumination Relighting</i> (QIR)	34
3.5	Método de Zhu et al. (2003)	37
3.5.1	Estimação da iluminação do fundo	38
3.5.2	Estimação de ganho	39
3.5.3	Normalização da imagem	40
3.6	Normalização da iluminação utilizando a Transformadas de <i>Wavelets</i>	41
3.7	Compensação de iluminação por meio de Redes Neurais Artificiais (RNAs) e <i>wavelets</i>	43
3.8	Variação do filtro homomórfico	45
3.9	Retinex.....	47
3.9.1	Retinex de Única Escala (SSR).....	48
3.9.2	Retinex de Múltipla Escala (MSR)	49
3.9.3	Retinex de Múltipla Escala com Restauração de Cor (MSRCR)	50
3.10	Considerações finais	51
4	Reconhecimento de Faces.....	53
4.1	Considerações iniciais	53
4.2	Métodos para o reconhecimento facial.....	53
4.3	Reconhecimento de faces utilizando PCA (Análise de Componentes Principais).....	55
4.3.1	Algoritmo de reconhecimento utilizado.....	57
4.4	Algoritmo de reconhecimento utilizando auto-faces.....	57
4.5	Reconhecimento facial utilizando rede neural MLP e <i>Wavelets</i>	58
4.6	Considerações finais	59
5	Metodologia	60
5.1	Considerações iniciais	60
5.2	Banco de imagens.....	60
5.3	Metodologia.....	62
5.3.1	Filtro LogAbout	63
5.3.2	Transformada de <i>Wavelet</i>	65
5.3.3	Filtro homomórfico	68

5.3.4	Tempo dos métodos de compensação da iluminação	69
6	Resultado e discussões	70
6.1	Considerações iniciais	70
6.2	Avaliação dos resultados através dos algoritmos de reconhecimento de faces..	70
6.2.1	Algoritmo utilizando PCA e rede neural LVQ	71
6.2.2	Algoritmo utilizando <i>wavelets</i> e rede neural	73
6.2.3	Algoritmo utilizando auto-faces	77
7	Conclusões e sugestões para trabalhos futuros	80
7.1	Considerações iniciais	80
7.2	Conclusões.....	80
7.3	Contribuições.....	81
7.4	Sugestões para trabalhos futuros	82
	Referências Bibliográficas	83
	Anexo A – Matriz de confusão com os resultados do algoritmo de reconhecimento facial PCA com LVQ	89
	Anexo B – Matriz de confusão com os resultados do algoritmo de reconhecimento facial <i>wavelet</i> com MLP	96
	Anexo C – Matriz de confusão com os resultados do algoritmo de reconhecimento facial auto-faces	103

1 Introdução

Variações de direção e de intensidade da iluminação são fatores que modificam significativamente a aparência dos objetos numa imagem digital. A iluminação ambiente pode variar ao longo do dia, em ambientes internos e externos. Devido à forma tridimensional dos objetos, uma fonte de iluminação pode gerar sombras que acentuam ou diminuem certos traços da imagem; além disso, condições distintas de iluminação podem produzir representações desiguais do mesmo objeto, dificultando o processo de segmentação e reconhecimento de padrões, incluindo o reconhecimento facial (ARAIZA et al., 2007).

As técnicas de realce de contraste melhoram a aparência global das imagens e têm como propósito final torná-las mais adequadas para uma aplicação específica ou processo determinado, como por exemplo, segmentação, detecção de contornos, reconhecimento de padrões, dentre outros. Mesmo com as diversas técnicas para realçar o contraste, o melhoramento das imagens afetadas por uma iluminação desigual continua sendo um problema de estudo, devido à natureza complexa dos objetos; e os efeitos produzidos sobre os mesmos, incluindo sombras e regiões muito claras (ARAIZA et al., 2007).

Para o caso particular de imagens de face, a questão de alteração da iluminação é crucial, pois pequenas mudanças na iluminação podem produzir grandes alterações na aparência da face, dificultando o seu reconhecimento.

Apesar dos diferentes métodos propostos para o reconhecimento facial, como PCA, Gabor, Redes Neurais Artificiais (KITANI e THOMAZ, 2006; LIU et al., 2004; GASPAR, 2006), problemas relacionados com a distribuição de iluminação para esse fim têm sido considerado um desafio na comunidade de visão computacional.

Vários trabalhos são encontrados na literatura com a finalidade de diminuir a má influência da iluminação, para posterior detecção facial e/ou reconhecimento, e alguns serão citados a seguir.

Algoritmos de realce de contraste bem conhecidos, como por exemplo, equalização do histograma e correção gama, são métodos globais e não consideram detalhes na imagem que são importantes para o reconhecimento facial (DU e WARD, 2005).

Liu et al. (2002) mencionam em seu trabalho as transformações logarítmicas e o filtro homomórfico para a compensação de iluminação. As transformações logarítmicas aumentam os baixos níveis de cinza e comprimem os altos, sendo úteis para imagens não uniformes e com sombras. Entretanto, não são úteis para imagens com brilho. O filtro homomórfico proporciona simultaneamente um intervalo dinâmico de compressão e realce de contraste. No entanto, este algoritmo é limitado ao tamanho da imagem $2^n \times 2^n$.

Shan et al. (2003) investigaram diversos métodos de normalização da iluminação e propuseram novas soluções, tais como: Correção de Intensidade Gama (GIC); Estratégia Baseada em Região combinando GIC e Equalização do Histograma (HE); e o método *Quotient Illumination Relighting* (QIR).

Zhu et al. (2003) propuseram um método para correção de iluminação normalizando o contraste da imagem por meio de uma transformação afim, efetuando uma estimação de fundo e ganho.

Chen et al. (2004) utilizaram a Transformada do Cosseno Discreta (DCT) para compensar variações da iluminação no domínio logarítmico. Este método não requer nenhum passo de modelagem, e pode ser facilmente implementado em sistema de reconhecimento facial em tempo real.

No trabalho de Zhang et al. (2005), são mencionados três métodos para amenizar a influência da iluminação.

O primeiro método é o de extração de características invariantes à iluminação, onde são extraídas características faciais que não se alteram ou não são sensíveis à variação de iluminação. Mas, Mose e Ullman (1992) e Adini et al. (1997) provaram teórica e experimentalmente que não é possível extrair características faciais que sejam insensíveis às mudanças de iluminação.

O segundo método estabelece um modelo para alterações de iluminação. A principal filosofia deste método é usar um subespaço próprio para representar as alterações de iluminação. O processo consiste em estimar parâmetros no modelo facial de teste e então escolher o subespaço mais próximo à face de teste.

O terceiro método faz a normalização da iluminação, diminuindo a influência de alterações de iluminação antes da imagem da face ser reconhecida, para em seguida, a face normalizada ser reconhecida.

1.1 Proposta

O desempenho da maioria dos sistemas de reconhecimento facial, tanto aqueles com resultados acadêmicos, como os comerciais, está fortemente sujeito às variações nas condições da imagem (ZHAO et al., 2003). Mudanças de posição e intensidade das fontes de luz contribuem para a grande variação da iluminação, como pode se observar na Figura 1, o que pode comprometer o desempenho do algoritmo de reconhecimento.

Figura 1: Imagens da mesma pessoa submetida a diferentes condições de iluminação.

FONTE: LEVINE et al., (2004).

Segundo Delac et al. (2006), variações entre imagens de diferentes faces podem ser menores do que variações entre imagens da mesma face, submetidas a diferentes condições de iluminação. Além disso, mudanças na iluminação podem causar maior variação em imagens de face do que mudanças de posição.

A proposta deste trabalho foi identificar um método adequado para melhoria da qualidade das imagens de faces no que se refere aos problemas de iluminação, por meio da comparação de métodos existentes para normalizá-las.

No trabalho desenvolvido, para demonstrar a eficiência do pré-processamento dos algoritmos de normalização de iluminação que foram comparados, as imagens de faces, após passarem por este pré-processamento, foram testadas em três algoritmos de reconhecimento facial que serão mencionados na próxima seção.

1.2 Objetivo

O objetivo foi o de identificar o método de normalização da iluminação que resulta na melhor taxa de reconhecimento facial. Para isso, foram utilizados os seguintes métodos encontrados na literatura, que apresentaram bons resultados para a normalização da iluminação de imagens de face: LogAbout (LIU et al., 2002), filtro homomórfico modificado (DELAC et al., 2006) e o método baseado em *wavelets* (DU e WARD, 2005).

Esses métodos foram aplicados sobre imagens faciais e foram avaliados por meio dos resultados obtidos com os seguintes algoritmos de reconhecimento facial: auto-faces (SERRANO, 2007), PCA (*Principal Component Analyses*) com rede neural LVQ (*Learning Vector Quantization*) (FERNANDE e GONZAGA, 2007) e *wavelets* com rede neural MLP (*Multilayer Perceptron*) (RIZK et al., 2004).

1.3 Estrutura do trabalho

Este primeiro capítulo apresentou a proposta e o objetivo deste trabalho, e os demais capítulos do texto estão organizados da seguinte forma:

- no segundo capítulo é apresentado o embasamento teórico do filtro homomórfico, da Transformada de *Wavelet* e Redes Neurais Artificiais, uma vez que, nesse trabalho, tais métodos são utilizados nos algoritmos de normalização da iluminação e reconhecimento facial;
- no terceiro capítulo são descritos métodos clássicos de realce de contraste, além de descrever detalhadamente alguns algoritmos de normalização da iluminação utilizados na literatura, para melhorar a taxa de detecção e/ou reconhecimento facial, e descrever os métodos utilizados neste trabalho;
- no quarto capítulo são descritos os algoritmos de reconhecimento facial utilizados neste trabalho;
- no quinto capítulo é apresentada a metodologia;
- no sexto capítulo são apresentadas as discussões e os resultados;
- no sétimo capítulo são apresentadas as conclusões e sugestões para trabalhos futuros;

2 Fundamentação teórica

2.1 Considerações iniciais

Este capítulo descreve a base teórica para uma melhor compreensão dos métodos de compensação da iluminação, que são utilizados neste projeto. São apresentados os conceitos de filtragem homomórfica, Transformada de *Wavelet*, e Redes Neurais Artificiais, dando-se ênfase às redes utilizadas nos algoritmos de reconhecimento facial deste projeto: rede MLP (*MultiLayer Perceptron*) e rede LVQ (*Learning Vector Quantization*).

2.2 Filtro Homomórfico

O filtro homomórfico é um filtro no domínio da frequência utilizado para realçar as altas frequências e atenuar as baixas frequências. Para sua melhor compreensão será apresentada uma breve introdução sobre filtros no domínio da frequência na seção 2.2.1, e os conceitos envolvidos em filtragem homomórfica na seção 2.2.2.

2.2.1 Filtros no domínio da frequência - conceitos

Para filtragem no domínio da frequência calcula-se a transformada de Fourier da imagem a ser realçada, multiplica-se o resultado por uma função de transferência, e determina-se a transformada inversa para produzir a imagem realçada (GONZALEZ e WOODS, 2002).

Os seguintes passos podem ser seguidos para aplicar a filtragem (GONZALEZ e WOODS, 2002):

1. Preencher imagem de entrada, $f(x,y)$, com zeros, de maneira que fique com o dobro do tamanho e seja múltiplo de 2.
2. Multiplicar a imagem de (1) por $(-1)^{x+y}$ para centralizar a transformada.
3. Computar $F(u,v)$, a DFT da imagem em (2).
4. Multiplicar $F(u,v)$ por um filtro $H(u,v)$.
5. Computar a DFT inversa do resultado (4).
6. Obter a parte real do resultado (5).
7. Multiplicar o resultado de (6) por $(-1)^{x+y}$.
8. Recortar região de interesse da imagem em (7).

O procedimento da filtragem mencionado anteriormente é apresentado na Figura 2 de uma forma mais geral, que inclui estágio de pré e pós-processamento.

Figura 2: Processamento no domínio da frequência.
FONTE: GONZALEZ e WOODS, (2002).

No processamento no domínio da frequência, além dos filtros passa-alta ideal e passa-baixa ideal existe outros filtros. Como exemplo pode-se citar o filtro Gaussiano e o filtro Butterworth, que podem ser passa-alta ou passa-baixa. O que os difere é a suavidade de transição na frequência de corte. A Figura 3 ilustra exemplos das curvas dos filtros passa-alta ideal, Butterworth passa-alta e Gaussiano passa-alta.

Figura 3: Curva dos filtros passa alta: a) Ideal. b) Butterworth. c) Gaussiano.

2.2.2 Filtragem Homomórfica

Filtragem homomórfica é um método de complexidade computacional moderada para compensar mudanças de iluminação, baseada no modelo de reflectância-iluminação. Nesse modelo, a iluminação $L(x, y)$ representa a quantidade de luz incidente na cena, e a reflectância $R(x, y)$, representa a quantidade de luz refletida pelos objetos na cena. Então, uma imagem $I(x, y)$ pode ser expressa como (GONZALEZ e WOODS, 2002; DELAC et al., 2003):

$$I(x, y) = R(x, y) \cdot L(x, y) \quad (1)$$

Esse modelo pode ser utilizado para a melhoria da qualidade de uma imagem que tenha sido adquirida sob condições de iluminação pobre.

O componente de iluminação $L(x, y)$ resulta das condições de luz presente quando a imagem é capturada e pode ser modificado quando as condições de iluminação são alteradas. A reflectância $R(x, y)$ resulta na maneira como os objetos na imagem refletem a luz, sendo determinada pelas propriedades intrínsecas do próprio objeto, o que não se altera. O componente de iluminação de uma imagem é geralmente caracterizado por variações espaciais lentas (baixa frequência espacial), enquanto o componente de reflectância tende a variar abruptamente (alta frequência espacial), particularmente na junção de objetos diferentes (DELAC et al., 2003).

A equação (1) não pode ser usada diretamente para operar separadamente os componentes de frequência da iluminação e reflectância, pois a transformada de Fourier do produto de duas funções não é separável, ou seja (GONZALEZ e WOODS, 2002):

$$\mathfrak{I}\{I(x, y)\} \neq \mathfrak{I}\{R(x, y)\} \cdot \mathfrak{I}\{L(x, y)\} \quad (2)$$

Supondo-se, entretanto, que seja definido

$$\begin{aligned} Z(x, y) &= \ln[I(x, y)] \\ &= \ln[R(x, y)] + \ln[L(x, y)] \end{aligned} \quad (3)$$

então,

$$\begin{aligned} \mathfrak{I}\{Z(x, y)\} &= \mathfrak{I}\{\ln[I(x, y)]\} \\ &= \mathfrak{I}\{\ln[R(x, y)]\} + \mathfrak{I}\{\ln[L(x, y)]\} \end{aligned} \quad (4)$$

ou

$$Z(u, v) = F_R(u, v) + F_L(u, v) \quad (5)$$

onde $F_R(u, v)$ e $F_L(u, v)$ são as Transformadas de Fourier de $\ln[R(x, y)]$ e $\ln[L(x, y)]$ respectivamente.

Se $Z(u, v)$ for processado por meio de uma função de filtro $H(u, v)$, obtém-se a versão filtrada $S(u, v)$.

$$\begin{aligned} S(u, v) &= H(u, v).Z(u, v) \\ &= H(u, v).F_R(u, v) + H(u, v).F_L(u, v) \end{aligned} \quad (6)$$

Para retornar ao domínio do espaço aplica-se a Transformada Discreta de Fourier Inversa obtém-se $s(x, y)$:

$$\begin{aligned} s(x, y) &= \mathfrak{I}^{-1}\{S(u, v)\} \\ &= \mathfrak{I}^{-1}\{H(u, v).F_R(u, v)\} + \mathfrak{I}^{-1}\{H(u, v).F_L(u, v)\} \end{aligned} \quad (7)$$

onde

$$L'(x, y) = \mathfrak{F}^{-1} \{H(u, v).F_L(u, v)\} \quad (8)$$

e

$$R'(x, y) = \mathfrak{F}^{-1} \{H(u, v).F_R(u, v)\} \quad (9)$$

A equação (7) pode ser expressa da seguinte forma:

$$s(x, y) = R'(x, y) + L'(x, y) \quad (10)$$

Finalmente, a imagem realçada é denotada por $I'(x, y)$.

$$\begin{aligned} I'(x, y) &= e^{s(x, y)} \\ &= e^{R'(x, y)} . e^{L'(x, y)} \\ &= R_0(x, y) L_0(x, y) \end{aligned} \quad (11)$$

onde

$$R_0(x, y) = e^{R'(x, y)} \quad (12)$$

e

$$L_0(x, y) = e^{L'(x, y)} \quad (13)$$

Um filtro passa-alta normalmente usado para este procedimento é o filtro Butterworth definido como (GONZALEZ E WOODS, 2002):

$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}} \quad (14)$$

onde n define a ordem do filtro, D_0 é a distância da frequência de corte do centro e $D(u, v)$ pode ser expresso pela equação (15):

$$D(u,v) = \left[(u - M/2)^2 + (v - N/2)^2 \right]^{1/2} \quad (15)$$

onde M é o número de linhas e N o número de colunas da imagem original.

Quanto maior a ordem do filtro, mais este se aproxima de um filtro passa-alta ideal. A Figura 4 ilustra a curva do filtro Butterworth passa-baixa.

Figura 4: Filtro Butterworth passa-baixa.
FONTE: GONZALEZ e WOODS, (2002).

A Figura 5 ilustra todos os passos da filtragem homomórfica, onde DFT é a Transformada Discreta de Fourier e $(DFT)^{-1}$ é sua transformada inversa. Um exemplo de aplicação da filtragem homomórfica é mostrado na Figura 6.

Figura 5: Passos filtro homomórfico.
FONTE: DELAC et al., (2003).

Figura 6: Aplicação do filtro Butterworth passa-alta: a) Imagem original. b) DFT. c) Imagem resultante com $D_0=0.5$ e $n=10$. d) Imagem resultante com $D_0=0.5$ e $n=25$.

2.3 Wavelets

Embora a Transformada de Fourier (FT) tenha sido a base principal do processamento de imagens baseada em transformada desde a década de 50, uma transformação mais recente denominada Transformada de *Wavelet* vem sendo usada para comprimir, transmitir, e analisar muitas imagens. Diferente da Transformada de Fourier cujas funções de base são senoidais, a Transformada de *Wavelet* baseia-se em pequenas ondas, denominadas *wavelets*, que variam em frequência e tem duração limitada (GONZALEZ e WOODS, 2002).

Uma das desvantagens da Transformada de Fourier é que a informação temporal é perdida no processo de transformação, o que não ocorre na Transformada de *Wavelet*. Para ilustrar, será considerada a Figura 7, que mostra dois tipos de sinais: estacionário (Figura 7 (a)), no qual a frequência não se altera no tempo, e sinal não-estacionário (Figura 7 (b)), onde a frequência se altera no tempo. A FT somente informa qual componente de frequência existe no sinal, porém a maioria dos sinais é não-estacionário, sendo necessário saber onde e quando uma determinada frequência ocorreu (QIAO, 2005).

Figura 7: Transformada de Fourier aplicada num sinal estacionário (a), e não-estacionário (b).
FONTE: QIAO, (2005).

Uma solução para este problema é a Transformada de Fourier Janelada (STFT), na qual janelas são utilizadas para analisar somente uma pequena parte do sinal no tempo. Porém, quando se usa uma janela de comprimento infinito, volta-se para a Transformada de Fourier, que oferece uma ótima resolução em frequência, mas sem informação temporal. Por outro lado, quando a janela é pequena obtém-se melhor resolução no tempo, porém pior é a resolução em frequência. Uma alternativa para a STFT é a Transformada de *Wavelet*, que analisa o sinal em diferentes frequências com diferentes resoluções (QIAO, 2005).

2.3.1 Transformada Discreta de *Wavelet*

Como já mencionado anteriormente, a Transformada de *Wavelet* utiliza janelas com comprimentos variáveis que se ajustam melhor às características do sinal em contraste com a STFT. A grande vantagem da Transformada de *Wavelet* é a possibilidade da análise tempo-frequência que permite a localização de altas frequências em intervalos de tempo específicos (SOARES, 2008).

Em sua versão discreta a Transformada de *Wavelet* é dada por:

$$f(t) = \sum_{k=-\infty}^{\infty} d_k \phi_k(t) + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} c_k^j \psi_k^j(t) \quad (16)$$

onde a função

$$\psi_k^j(t) = \psi(2^j t - k) \quad (17)$$

é chamada de *wavelet* mãe e a função

$$\phi_{j,k}(t) = 2^{\frac{j}{2}} \phi(2^j t - k) \quad (18)$$

é chamada de *wavelet* pai ou função escala e os coeficientes j e k são os coeficientes de transformação *wavelet*.

A transformada *wavelet* especifica a localização (via translação) e frequência (via dilatação ou compressão) utilizando a *wavelet* mãe e pai respectivamente.

$$\psi_{c,b}(t) = \frac{1}{\sqrt{c}} \psi\left(\frac{t-b}{c}\right) \quad (19)$$

$$\phi_{c,b}(t) = \frac{1}{\sqrt{c}} \phi\left(\frac{t-b}{c}\right) \quad (20)$$

onde os parâmetros c e b variam continuamente no espaço em R , de modo a obter a *wavelet* dilatada ou comprimida pelo fator c e deslocada pelo fator b . Através de compressões/dilatações e deslocamentos das funções *wavelets* é possível aproximar qualquer sinal nos planos R e I .

Para $c > 1$ têm-se o efeito de dilatação da função *wavelet* e para $c < 1$ têm-se o efeito de contração da função *wavelet* sendo possível analisar os aspectos globais e locais das funções. À medida que b varia, a função f é analisada localmente em torno de b .

2.3.2 Bases Wavelets

Enquanto a Transformada de Fourier utiliza funções trigonométricas seno e cosseno, a transformada *wavelet* pode utilizar qualquer função matemática que satisfaça os critérios de admissibilidade expressos nas equações (21) e (22) (SOARES, 2008).

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (21)$$

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt = 1 \quad (22)$$

A equação (21) expressa que para uma função ser *wavelet*, sua integral deve ser zero. Isso garante que a função *wavelet* tenha uma forma do tipo onda, e a equação (22) define que a função deve ter energia unitária, garantindo que a função tenha decaimento rápido para a localização temporal (HERNANDEZ e WEISS, 1996; MALLAT, 1999).

A *wavelet* de Haar foi a primeira função conhecida com comportamento *wavelet* e também a mais simples em termos matemáticos. A *wavelet* de Haar é dada pela equação (23).

$$\psi(t) = \begin{cases} 1 & t \leq \frac{1}{2} \\ -1 & \frac{1}{2} \leq t < 1 \\ 0 & t > 1 \end{cases} \quad (23)$$

A base de Haar tem o inconveniente de ser formada por funções descontínuas, expressas na equação (23), entretanto sua facilidade de compreensão e implementação motivam o uso da base de Haar como *wavelet* mãe.

A partir da base de Haar outras bases foram desenvolvidas na forma de família de funções similares, como por exemplo a família de bases de Daubechies (DAUBECHIES, 1990), Meyer (MEYER, 1993) e Coifet (HUANG e HSIEH, 2002). A Figura 8 ilustra a representação dessas bases.

Figura 8: Exemplo de funções base *wavelets*.
FONTE: SOARES, (2008).

Uma questão que sempre surge na aplicação da análise *wavelet* é a escolha da base *wavelet* mais adequada. A escolha da base *wavelet* é um fator importante, mas a escolha de uma base adequada ainda é tema de pesquisa em aberto. Uma alternativa é definir experimentalmente a base *wavelet* a ser utilizada no problema em estudo.

2.3.3 Análise Multiresolução

A análise de *wavelet* consiste em aproximações e detalhes. As aproximações representam os componentes de baixa frequência do sinal. Os detalhes são as componentes de alta frequência do sinal. A representação de um sinal em diferentes níveis de detalhamento é conhecida como multiresolução.

Como exemplo, a Figura 9 mostra a Árvore de Decomposição *Wavelet* até o terceiro nível, para um sinal S , decomposto em coeficientes de aproximação $A1$ e coeficientes de detalhes $D1$ do nível 1. A aproximação do nível 1 $A1$, é também decomposta em coeficientes de aproximação $A2$ e coeficientes de detalhes $D2$ do nível 2. Este processo ocorre, de forma recursiva, até o n -ésimo nível de decomposição (REIGOTA, 2007).

Figura 9: Árvore de Decomposição Wavelet.
FONTE: Toolbox do Matlab 07.

A análise de *wavelet* aplicada à filtragem de imagens pode ser implementada a partir de um Banco de Filtros, denominados de filtros em Quadratura Conjugada (QMF – *Quadrature Mirror Filters*).

Os coeficientes do filtro passa-alta (H) são determinados pela função *wavelet* ψ ou função mãe, enquanto os coeficientes do filtro passa baixa (L) são determinados pela função Φ ou função escala.

De forma resumida, pode-se entender que o processo de decomposição em multiresolução consiste em separar um sinal original em duas partes. Uma parte contém as componentes de grande escala do sinal e a outra contém as componentes de pequena escala representadas no diagrama da Figura 10. Esse diagrama consiste de um conjunto de filtros de Análise e Síntese, responsáveis pela decomposição e reconstrução da imagem. A letra **H** representa o filtro passa-alta e a letra **L** representa o filtro passa-baixa. Eles são seguidos por um estágio de sub-amostragem para geração dos coeficientes de aproximação e de detalhes horizontais, verticais e diagonais (REIGOTA, 2007).

O processo de reconstrução da imagem ou do sinal, a partir dos coeficientes de aproximação **cA** e detalhes **cD**, é obtido após o estágio de sub-amostragem, seguido por filtros passa-alta **H'** e passa-baixa **L'** de síntese.

Figura 10: Banco de filtros.
FONTE: Toolbox do Matlab 07.

O processo de decomposição é recursivo, isto é, aplica-se tantas vezes quanto necessário, para obter os coeficientes de aproximação e detalhe de um sinal ou imagem.

2.3.4 Transformada *Wavelet* bi-dimensional

A transformada *Wavelet* bi-dimensional para sinais discretos da função $f(x,y)$ de tamanho $M \times N$ é definida como (GONZALEZ e WOODS, 2002):

$$W_{\Phi}(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \Phi_{j_0, m, n}(x, y) \quad (24)$$

$$W_{\Psi}^j(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \Psi_{j, m, n}^i(x, y) \quad (25)$$

onde $i = \{H, V, D\}$, j_0 é uma escala inicial arbitrária, e os coeficientes $W_{\Phi}(j_0, m, n)$ definem a aproximação de $f(x, y)$ na escala j_0 . Os coeficientes $W_{\Psi}^j(j, m, n)$ adicionam detalhes horizontal, vertical e diagonal para as escalas $j \geq j_0$. Normalmente, $j_0 = 0$ e $N = M = 2^J$ de forma que $j=0, 1, 2, \dots, J-1$ e $m, n=0, 1, 2, \dots, 2^j - 1$.

Para a Transformada de *Wavelet* Discreta (DWT), uma imagem é representada em termos de translações e dilatações, das funções escala e *wavelet*. Os coeficientes de escala e da *wavelet* podem ser facilmente calculados utilizando um banco de filtros 2D, consistindo de

filtros passa-alta e passa-baixa. Após um nível de decomposição 2D, a imagem é dividida em quatro sub-bandas (Figura 11) (DU e WARD, 2005):

- LL: que é gerada pelos coeficientes de aproximação;
- LH: que é gerado pelos coeficientes de detalhe horizontal;
- HL: que é gerado pelos coeficientes de detalhe vertical
- HH: que é gerado pelos coeficientes de detalhe diagonal.

Figura 11: Esquema de decomposição bi-dimensional

A sub-banda LL contém uma aproximação da imagem original em uma resolução equivalente à metade da imagem original. É obtida aplicando-se o filtro passa-baixa nas linhas e colunas. De maneira análoga, a sub-banda HH é obtida aplicando-se o filtro passa-alta nas linhas e colunas. HL é o resultado de filtragem passa-alta nas linhas seguida de filtragem passa-baixa nas colunas. Do mesmo modo, LH é resultado de filtragem passa-baixa nas linhas, seguida da filtragem passa-alta nas colunas, conforme pode ser observado na Figura 12 (LUCCA, 2006).

Figura 12: Decomposição de uma imagem. PA e PB indicam filtros passa-alta e passa-baixa, respectivamente. A componente LL é utilizada como dado de entrada para o próximo nível. Esta arquitetura de filtragem é denominada de banco de filtros.

FONTE: LUCCA, (2006)

2.3.5 Wavelets de Daubechies

Entre as diferentes *wavelets* conhecidas será apresentada a seguir aquela que foi utilizada no algoritmo de normalização de iluminação deste trabalho: as *wavelets* de *Daubechies*.

As *wavelets* de *Daubechies* possuem maior regularidade (suavidade) e apresentam uma capacidade de análise e síntese mais efetiva do que a *wavelet* de Haar. São definidas num determinado intervalo de tempo, motivo pelo qual são chamadas geralmente de *wavelets* compactamente definidas no domínio do tempo (FARIA, 1997).

Elas são numeradas, onde o índice corresponde ao número de coeficientes que os filtros associados possuem, que é também o número de derivadas nulas do filtro associado à função escaladora em $\omega = \pi$ (o filtro tem um zero em $\omega = \pi$ de ordem $N-1$). Assim, a *wavelet*

Daub 4 possui 2 momentos nulos, e os filtros associados possuem 4 zeros em $\omega = \pi$, e 4 coeficientes (REIGOTA, 2007). A *wavelet* utilizada foi a Daub 10.

A *wavelet* de Daubechies utilizada neste trabalho, foi a Daub 10, que possui os seguintes coeficientes:

- $C_0 = 0.160102397$
- $C_1 = 0.603829269$
- $C_2 = 0.724308528$
- $C_3 = 0.138428145$
- $C_4 = -0.242294887$
- $C_5 = -0.032244869$
- $C_6 = 0.077571493$
- $C_7 = -0.006241490$
- $C_8 = -0.001258075$
- $C_9 = 0.003335725$

Uma vez que as *wavelets* podem ser representadas como banco de filtros, a operação entre esses bancos será apresentada a seguir na forma matricial por meio de uma multiplicação de matrizes, onde uma delas é a matriz de coeficientes da *wavelet* de Daubechie escolhida.

Para transformar uma seqüência N nos seus N coeficientes (gerados pela decomposição) é necessário uma matriz $N \times N$. A transformada inversa envolve o uso da matriz inversa. Considerando x um vetor de entrada, b o vetor transformado (com N coeficientes *wavelet*), W a matriz de transformação *wavelet* montada com os coeficientes dos filtros do banco ($h(n)$ e $g(n)$). A Análise e Síntese, responsáveis pela decomposição e reconstrução da imagem, respectivamente, pode ser representada pelas equações (FARIA, 1997):

$$\begin{aligned} \text{Análise: } b &= W.x \\ \text{Síntese: } x &= W^{-1}.b \end{aligned} \tag{26}$$

A escolha do filtro passa-alta e passa-baixa (equivalente à escolha da base *wavelet*) executará uma forte influência nas propriedades verificadas pelas matrizes. Por exemplo, sendo $h[]$ e $g[]$ ortogonais, o banco de filtros será ortogonal e a matriz correspondente também será ortogonal. Por meio de uma normalização a ortogonalidade passa a ortonormalidade, e tem-se que:

$$W^T.W = I \Rightarrow W^{-1} = W^T$$

ou seja, o banco de síntese é o transposto do banco de análise. Esta condição permite que as matrizes possam ser fatoradas em matrizes menores e mais esparsas, gerando desta forma, uma transformada rápida, e diminuindo o número de operações necessárias para a transformação *wavelet* (FARIA, 1997).

Para ilustrar, será utilizada a *wavelet* Daub 4, cuja matriz está representada na Figura 13.

$$\begin{bmatrix} c_0 & c_1 & c_2 & c_3 & & & & & & & \\ c_3 & -c_2 & c_1 & -c_0 & & & & & & & \\ & & c_0 & c_1 & c_2 & c_3 & & & & & \\ & & c_3 & -c_2 & c_1 & -c_0 & & & & & \\ \vdots & \vdots & & & & & \ddots & & & & \\ & & & & & & & c_0 & c_1 & c_2 & c_3 \\ & & & & & & & c_3 & -c_2 & c_1 & -c_0 \\ & & & & & & & & & c_0 & c_1 \\ c_2 & c_3 & & & & & & & & c_3 & -c_2 \\ c_1 & -c_0 & & & & & & & & & \end{bmatrix}$$

Figura 13: Matriz de coeficientes da Daub 4.
FONTE: VIEIRA (2006).

Onde os valores dos coeficientes são:

- $C_0 = 0.482962$;
- $C_1 = 0.8365163$;
- $C_2 = 0.2241138$;
- $C_3 = -0.1294095$

Na matriz de coeficientes, as entradas em branco correspondem aos valores zero. A primeira, terceira, quinta, e as outras linhas ímpares geram componentes dos dados convoluídos com o filtro dos coeficientes $[c_0, c_1, c_2, c_3]$. Ao invés das linhas pares convoluírem com os coeficientes do filtro, estas fazem uma convolução diferente com $[c_3, -c_2, c_1, -c_0]$ (VIEIRA, 2006). O filtro $[c_0, c_1, c_2, c_3]$ é o filtro passa-baixa ou de suavização $h[]$, enquanto que $[c_3, -c_2, c_1, -c_0]$ é o filtro passa-alta $g[]$.

Uma vez que os coeficientes da matriz da *wavelet* de Daubechie é ortogonal, sua inversa é a matriz transposta conforme está ilustrado na Figura 14.

$$\begin{bmatrix} c_0 & c_3 & & \dots & & & c_2 & c_1 \\ c_1 & -c_2 & & \dots & & & c_3 & -c_0 \\ c_2 & c_1 & c_0 & c_3 & & & & \\ c_3 & -c_0 & c_1 & -c_2 & & & & \\ & & & \ddots & & & & \\ & & & & c_2 & c_1 & c_0 & c_3 \\ & & & & c_3 & -c_0 & c_1 & -c_2 \\ & & & & & c_2 & c_1 & c_0 & c_3 \\ & & & & & c_3 & -c_0 & c_1 & -c_2 \end{bmatrix}$$

Figura 14: Matriz inversa.
FONTE: VIEIRA, 2006.

2.4 Redes Neurais Artificiais

Redes Neurais Artificiais (RNAs), também conhecidas como connexionismo ou sistemas de processamento paralelo e distribuídos, são modelos computacionais inspirados na estrutura neural de organismos inteligentes e que adquirem conhecimento por meio da experiência (BRAGA et al., 1998).

Segundo Haykin (2001), uma RNA é um processador paralelamente distribuído composto por unidades de processamento simples, que tem capacidade natural em armazenar conhecimento experimental e torná-lo disponível para uso, assemelhando-se ao cérebro humano, pois o conhecimento é adquirido a partir de um processo de aprendizagem e forças de conexão entre neurônios, conhecidas como pesos sinápticos, que são utilizados para armazenar o conhecimento adquirido.

No modelo computacional, ilustrado na Figura 15, um neurônio é uma unidade de processamento fundamental para a rede neural, sendo possível identificar três elementos básicos nesse modelo: conjunto de entradas, o somador e a função de ativação.

Figura 15: Modelo de um neurônio artificial.
FONTE: HAYKIN, (2001).

Na Figura 15, os elos de conexão ou sinapse de um neurônio são caracterizados por um peso sináptico w_{kj} . Os terminais de entrada x_j apresentados à rede são multiplicados por w_{kj} , onde k é o índice do neurônio e j é o terminal de entrada da sinapse. O somador, ou combinador linear, é responsável pela soma dos sinais de entrada ponderados pelas respectivas sinapses. A função de ativação restringe o valor de saída do neurônio, ou seja, processa o conjunto de entradas recebidas e o transforma em estado de ativação.

Existem diversas funções de ativações. A rede Perceptron de Múltiplas Camadas (MLP), que será explicada na seção 2.4.2, tem um número limitado de funções de ativação, pois neste tipo de rede, as saídas especificam limites além da função de ativação diferenciável (BAILEY, 2006). A Figura 16 mostra as funções de ativação mais comuns nesse tipo de rede.

Figura 16: Funções de ativação comuns.
FONTE: BAILEY, (2006).

Os sistemas baseados em RNAs dependem fortemente da arquitetura destas redes (tamanho, estrutura, conexões). As principais arquiteturas de uma rede neural são: *feedforward*, recorrentes e estrutura reticulada (CASTRO, 2003).

Como os algoritmos de reconhecimento de faces desse trabalho usam arquitetura *feedforward*, será abordada essa arquitetura, que pode ser de dois tipos (Figura 17):

-*Feedforward* de camada única: possui apenas a camada de entrada e saída.

-*Feedforward* de múltiplas camadas: além da camada de entrada e saída, possui uma ou mais camadas ocultas.

Figura 17: Arquitetura *feedforward*: a) De camada única. b) De múltiplas camadas.
FONTE: CASTRO, (2003).

2.4.1 Aprendizado Supervisionado

Uma propriedade importante das redes neurais é sua habilidade de aprender e melhorar seu desempenho por meio da aprendizagem, ou seja, através de um processo iterativo de ajustes aplicados a seus pesos (treinamento).

Existem vários algoritmos para o treinamento da rede, e eles podem ser agrupados em dois paradigmas principais: aprendizado supervisionado e aprendizado não supervisionado.

No aprendizado supervisionado, ilustrado na Figura 18, a entrada e a saída desejada para a rede são fornecidas por um supervisor (professor) externo que indica um comportamento para a rede, com o objetivo de direcionar o processo de treinamento, ajustar os parâmetros da rede a cada etapa do treinamento, de modo que se caminhe para uma solução, caso exista uma solução possível (BRAGA et al., 1998).

Figura 18: Aprendizado supervisionado de uma rede neural.
FONTE: BRAGA et al., (1998).

No aprendizado não supervisionado, não há exemplos rotulados da função a ser apreendida pela rede. A rede adapta seus dados de entrada estatisticamente, desenvolvendo uma habilidade para criar representações internas, que codificam as características dos dados de entrada e gera novas classes automaticamente (GASPAR, 2006).

2.4.2 Redes Perceptron de Múltiplas Camadas (MLP)

As Redes Perceptron de Múltiplas Camadas (MLP) são redes constituídas pela camada de entrada, uma ou mais camadas ocultas e pela camada de saída, conforme ilustrado na Figura 17 (b). As principais aplicações desta rede incluem: reconhecimento de padrões, aproximador universal de funções, processamento de imagens e de sinais, e previsão de séries temporais.

O *backpropagation* é um dos algoritmos supervisionado para efetuar o treinamento de uma rede MLP. Ele utiliza pares (entrada, saída desejada) para, através de um mecanismo de correção de erros, ajustar os pesos da rede. O treinamento ocorre em duas fases, onde cada fase percorre a rede em um sentido. Estas duas fases são chamadas de fase *forward* e fase *backward* (BRAGA et al., 1998).

Na fase *forward*, o vetor de entrada que é apresentado à rede propaga-se camada a camada. Durante este processo de propagação, os pesos sinápticos da rede não se alteram. Os pesos são ajustados na fase *backward* de acordo com a regra de correção de erro (regra delta). Os pesos são ajustados de maneira que a resposta gerada aproxima-se da resposta desejada. Este processo é repetido para todo conjunto de treinamento até que o erro seja menor do que o especificado (HAYKIN, 2001).

O objetivo do processo de treinamento é minimizar o sinal do erro entre a saída desejada e a resposta atual produzida pela rede. O sinal de erro da saída do neurônio y_i na interação n , é dado por:

$$e_i(n) = d_i(n) - y_i(n) \quad (27)$$

onde d_i é a saída desejada do i -th neurônio de saída.

Adicionando todos os erros quadráticos produzidos pelos neurônios de saída da rede com a i -th respectiva interação, obtêm-se a seguinte função de custo ou erro para ser minimizada:

$$E(w) = \frac{1}{2} \sum_i e_i^2 = \frac{1}{2} \sum_i (d_i - y_i)^2 \quad (28)$$

A função de custo é uma função continuamente diferenciada para cada peso, onde um algoritmo do gradiente descendente pode ser usado para ajustar os pesos. Para as conexões w_{ij} , que ligam a camada oculta e a camada de saída, a regra do gradiente descendente é dada por:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = \eta \delta_i \cdot v_j \quad (29)$$

onde η é a taxa de aprendizagem do algoritmo *backpropagation* e δ_i é o gradiente local definido por:

$$\delta_i = \phi'(c_i) \cdot (d_i - y_i) \quad (30)$$

onde c_i é o sinal de entrada dos pesos, aplicado para o neurônio y_i e $\phi(.)$ é a função de ativação do neurônio. As conexões w_{ij} são atualizadas iterativamente aplicando a seguinte equação:

$$w_{ij}^{novo} = w_{ij}^{velho} + \Delta w_{ij} \quad (31)$$

Para as conexões w_{jk} , que ligam a camada de entrada e a camada oculta, a função de custo é diferenciada com o respectivo w_{jk} usando a regra da cadeia, que é dada por:

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} = -\eta \cdot \sum \frac{\partial e}{\partial v_j} \cdot \frac{\partial v_j}{\partial w_{jk}} = \eta \cdot \delta_j \cdot x_k \quad (32)$$

onde o gradiente local δ_j em relação a w_{jk} é expresso por:

$$\delta_j = \phi'(h_j) \sum w_{ij} \cdot \delta_i \quad (33)$$

onde h_i é o peso do sinal de entrada aplicado para o neurônio v_j . Finalmente, todas as conexões w_{jk} são atualizadas iterativamente de acordo com a equação dada a seguir:

$$w_{jk}^{novo} = w_{jk}^{velho} + \Delta w_{jk} \quad (34)$$

Algumas técnicas auxiliam a melhorar o desempenho do treinamento da MLP (GASPAR, 2006):

- **Validação cruzada:** utilizada para testar o poder de generalização de uma MLP. Consiste em particionar o conjunto de dados em dois subconjuntos: conjunto de treinamento, utilizado para treinar a rede (80% a 90% do total dos dados), e conjunto de teste, utilizado para avaliar se a rede está generalizando de forma satisfatória (10% a 20% do total dos dados).
 - O mesmo conjunto de treinamento é utilizado para ajustar todas as topologias candidatas e, após o treinamento o conjunto de teste é utilizado para avaliar e escolher (validar) o melhor modelo (melhor generalização).
 - Em certos casos a rede pode convergir com os dados de treinamento, porém durante a fase de teste/validação, a rede produz resultados insatisfatórios;

- O aumento de neurônios e/ou camadas não significa que a rede irá generalizar melhor.
- **Normalização dos dados:** as variáveis referentes às entradas da rede devem ser normalizadas para a faixa $[0,1]$, se estiver sendo utilizada a função sigmóide, ou então, para $[-1,1]$ se for utilizada a tangente hiperbólica.

2.4.2.1 Treinamento

Para a solução de um determinado problema, a rede passa inicialmente por uma fase treinamento, e extrai características relevantes, criando uma representação própria para o problema. O treinamento tem como objetivo ajustar os pesos da rede de acordo com padrões apresentados, ou seja, elas aprendem por meio de exemplos (CARVALHO, 2007).

O treinamento de uma rede MLP utilizando *backpropagation* consiste em dois passos (CARVALHO, 2007):

- Calcular a resposta que é produzida na camada de saída (o erro é calculado) a partir do padrão que é apresentado às unidades da camada de entrada.
- O erro é propagado a partir da camada de saída até a camada de entrada, e, através da regra delta, os pesos das conexões das unidades das camadas internas vão sendo modificados.

2.4.3 *Learning Vector Quantization* (LVQ)

A rede LVQ é um classificador supervisionado que divide o espaço de dados dentro de regiões quantizadoras que limitam hiper-planos. Estas regiões são equivalentes a conjuntos de Voronoi. Essa rede é composta por duas camadas, sendo uma camada competitiva e uma camada de saída, Figura 19 (a). A camada competitiva aprende a classificar vetores de entrada, e a camada linear (camada de saída) transforma as classes da camada competitiva nas classificações definidas pelo usuário. As classes formadas pela camada competitiva são chamadas de subclasses e as classes da camada linear são chamadas de classes alvo (STRICKERT, 2004).

Figura 19: a) Rede LVQ. b) Diagrama de Voronoi.
FONTE: STRICKERT, (2004).

O método de aprendizagem da rede LVQ é chamado de aprendizado competitivo, e para cada padrão de treinamento, é determinado o vetor de referência que é o mais próximo a ele. O neurônio correspondente de saída é também conhecido como o neurônio vencedor. Para cada padrão de entrada apenas um neurônio de saída torna-se ativo. E, os pesos do neurônio vencedor são adaptados. Esta situação é descrita como “*winner-takes-all*”. A direção da adaptação depende da classe do padrão de treinamento e da classe atribuída ou não ao vetor coincidente. Se coincidirem, o vetor de referência é movido para mais perto do padrão de treinamento; caso contrário é afastado. Este movimento do vetor de referência é controlado por um parâmetro chamado de taxa de aprendizagem, que indica a fração de distância para o padrão do treinamento, e o quão distante o vetor de referência deve ser movido. O aprendizado pode ser finalizado quando as posições dos vetores de referência não mudam significativamente (THULASIRAMAN, 2005).

O algoritmo da rede LVQ é apresentado na Tabela 1.

Tabela 1: Algoritmo de treinamento da rede LVQ.

Algoritmo LVQ	
Iniciar vetor de pesos w aleatoriamente	
Repita	
Para todo padrão de entrada $x_i \in w_j; \quad j=1,2,...N$	
$k \leftarrow \arg \min \{d(x_i, w_j)\}$	{neurônio com menor distância d }
Se $x_i \in Classe_j$	
Então $w_j \leftarrow w_j + \gamma \cdot (x_i - w_j)$	{mesma classe}
Senão $w_j \leftarrow w_j - \gamma \cdot (x_i - w_j)$	{classe diferente}
Fim se	
Até que não ocorram mudanças em w	

FONTE: Adaptado STRICKERT (2004).

2.5 Considerações finais

Neste capítulo foi introduzida a teoria necessária para a compreensão dos seguintes métodos aplicados para normalização de iluminação e que são utilizados neste trabalho: filtro homomórfico adaptativo (DELAC et al., 2006) e *wavelets* (DU e WARD, 2005). Além disso, foi apresentada a fundamentação teórica das RNAs que fazem parte dos algoritmos de reconhecimento facial utilizados neste trabalho.

3 Métodos para correção de iluminação

3.1 Considerações iniciais

Neste capítulo serão apresentados alguns métodos clássicos de realce de contraste, e os trabalhos encontrados na literatura sobre a amenização da má influência da iluminação na detecção e/ou reconhecimento facial.

3.2 Realce de imagens no domínio do espaço

O objetivo principal do realce de imagem é processar uma imagem de modo que o resultado seja mais apropriado para uma aplicação específica do que a imagem original.

Realçar uma imagem f significa transformá-la em uma imagem g utilizando um operador T definido em uma vizinhança (x,y) . Os valores dos *pixels* das imagens f e g são denotados por r e s , conforme a equação (35) (GONZALEZ e WOODS, 2002).

$$s = T(r) \tag{35}$$

Dentre as técnicas de realce, pode-se citar as seguintes transformações nos níveis de cinza: negativo de uma imagem, transformação logarítmica, transformação gama e transformação de potência. Além das técnicas mencionadas de realce de imagens, ainda existem os operadores lógicos, processamento de histograma, filtros de suavização e filtros de aguçamento.

Será apresentado a seguir a transformação logarítmica (seção 3.2.1) e o método de equalização de histograma (seção 3.2.2).

3.2.1 Transformações nos níveis de cinza

As transformações nos níveis de cinza aumentam o contraste de faixas de intensidades ou até mesmo binariza as imagens. As transformações mais usadas são: transformação logarítmica, logarítmica inversa, correção, imagem negativa, raiz e potência. A Figura 20 ilustra a curva de cada uma destas transformações.

Figura 20: Transformações não-lineares.

FONTE: GONZALEZ e WOODS, (2002).

No método LogAbout de Liu et al. (2002), seção 3.3, é utilizada a transformação logarítmica, sendo essa transformação denotada por:

$$g = c \cdot \log(a + 1) \quad (36)$$

onde a é a imagem original, c é uma constante e g é a imagem final.

Esta função mapeia uma pequena faixa de pequenos valores de entrada em uma faixa maior de valores de saída, aumentando o contraste das áreas com baixas intensidades e

diminuindo o contraste das áreas com altas intensidades. Transformações logarítmicas são úteis em imagens com sombra e iluminação não-uniforme (LIU et al., 2002).

3.2.2 Equalização do Histograma (HE) e Equalização Adaptativa de Histograma (AHE)

O objetivo da HE (Equalização do Histograma) é redistribuir os valores de tons de cinza dos *pixels* em uma imagem, de modo a obter um histograma mais uniforme. Para tanto, utiliza-se uma função auxiliar denominada função de transformação. A forma mais usual de se equalizar um histograma é utilizar a Função de Distribuição Acumulada (CDF – *Cumulative Distribution Function*) da distribuição de probabilidades, que pode ser expressa por (MARQUES FILHO e VIEIRA NETO, 1999):

$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k P_i(r_j) \quad (37)$$

onde:

n é o número total de *pixels* da imagem,

$$0 \leq r_k \leq 1$$

$K = 0, 1, \dots, L-1$, onde L é o número de níveis de cinza da imagem digitalizada,

$P_i(r_j)$ é a probabilidade do j -ésimo nível de cinza.

A equalização do histograma é um método global utilizado para visualizar a imagem mais claramente, mas a HE produz um resultado não satisfatório em algumas regiões locais. A Equalização Adaptativa de Histograma (AHE) particiona a imagem em sub-regiões $N \times N$ e o histograma para cada região é calculado, ou seja, consiste em realçar o contraste local da imagem operando em regiões pequenas. Métodos de Equalização Adaptativa de Histograma e suas variações são descritos em PIZER et al. (1987). Esse método possui um alto custo computacional (ZHU et al., 2003).

3.3 Método LogAbout

O método LogAbout proposto por Liu et al. (2002) tenta resolver problemas de compensação de iluminação em detecção de face. Este método consiste na aplicação de um filtro passa-alta na imagem original, seguida de uma transformação logarítmica (Figura 21). Algumas vezes é necessário usar um filtro passa-baixa após utilizar o LogAbout, para reduzir falsas bordas e ruído.

Figura 21: Método LogAbout.

FONTE: Adaptado de LIU et al., (2002).

Os resultados dos experimentos de Liu et al. (2002), ilustrados na Figura 22, mostram que o método LogAbout pode compensar a iluminação eficientemente até mesmo em imagens com fundos complexos, porém não é o melhor para contraste local e condição de iluminação local.

Figura 22: Normalização com o método LogAbout: (1) Imagem original. (2) Imagem resultante da transformação logarítmica. (3) Imagem resultante.

FONTE: LIU et al., (2002)

3.4 Correção de Intensidade Gama (GIC), Estratégia Baseada em Região e *Quotient Illumination Relighting* (QIR)

Os métodos GIC, Estratégia Baseada em Região e QIR são propostos por Shan et al. (2003) com a finalidade de efetuar uma comparação entre métodos de normalização de iluminação.

O método GIC, diferente do método de Correção Gama da computação gráfica, consiste em normalizar toda intensidade da imagem para um dado nível de intensidade. Dada uma imagem de face I , capturada sob condições de luz desconhecidas, a imagem normalizada é computada por uma transformação Gama I'_{xy} dada por (Shan et al., 2003):

$$I'_{xy} = G(I_{x,y}; \gamma^*) \quad (38)$$

onde o coeficiente Gama γ^* é calculado conforme a equação (39), e visa minimizar a diferença entre a imagem transformada e a imagem pré-definida da face normal I_0 . $I_{x,y}$ é o nível de cinza da imagem na posição (x,y) .

$$\gamma^* = \arg \min_{\gamma} \sum_{x,y} [G(I_{x,y}; \gamma) - I_0(x,y)]^2 \quad (39)$$

A transformação gama é dada por:

$$G(I_{x,y}; \gamma) = c \cdot I_{xy}^{1/\gamma} \quad (40)$$

onde c é o parâmetro de alargamento do nível de cinza e γ é o coeficiente Gama.

HE e GIC são transformações globais sobre toda área da imagem, apresentando resultados não satisfatórios quando existem luzes laterais. Para resolver parcialmente este problema e diminuir efeitos causados por iluminação não uniforme, como por exemplo, sombra, a imagem é dividida em quatro regiões (Figura 23), e HE ou GIC é aplicada nas quatro regiões separadamente. A Estratégia Baseada em Região utilizando HE foi abreviada para RHE, e a baseada em GIC foi abreviada para RGIC.

Figura 23: Imagem dividida em quatro regiões para normalização da iluminação.
FONTE: SHAN et al., (2003).

No método QIR, o objetivo de Shan et al. (2003) foi iluminar novamente as imagens sob condição de iluminação pré-definida, denominada forma canônica, a partir de imagens de face obtidas sob condição de iluminação não normal (imagens com sombras em diferentes posições).

Uma vez que a condição de iluminação de uma imagem é determinada, é possível utilizar essa informação para realizar a normalização da iluminação, onde o procedimento é descrito a seguir (MITRA e SAVVIDES, 2005).

Dada uma imagem de face arbitrária, $I_{ij}(x, y)$, e a imagem canônica com a 0-ésima iluminação normal pré-definida, assume-se que a imagem é iluminada pela j -ésima condição de iluminação conhecida. Então, o j -ésimo quociente da iluminação é calculado como (MITRA e SAVVIDES, 2005):

$$R_j(x, y) = \frac{I_{ij}(x, y)}{I_{i0}(x, y)} \quad (41)$$

E a imagem canônica de I_{ij} , sob condição de iluminação canônica, pode ser calculada como:

$$I_{ij}^0(x, y) = \frac{I_{ij}(x, y)}{R_j(x, y)} \quad (42)$$

Isto proporciona uma maneira simples e direta para a normalização da iluminação, uma vez que a direção da fonte de luz da imagem pode ser conhecida. Considerando as imagens normalizadas I_{ij}^0 , qualquer método de identificação que não seja robusto às variações de iluminação pode agora ser aplicado sem dificuldades (Mitra e Savvides, 2005).

Experimentos de Shan et al. (2003) mostraram interessantes aspectos a respeito dos métodos comparados (ilustrados na Figura 24):

- métodos simples de normalização de iluminação, como por exemplo, a EH ou GIC, podem melhorar a taxa de reconhecimento de faces comparado com a não utilização de um método para esta finalidade;
- RHE e/ou GIC também podem melhorar significativamente a taxa de reconhecimento, pois esses métodos eliminam efeitos consideráveis de luzes laterais;
- se o modelo de iluminação da imagem de entrada for conhecido ou pode ser estimado, o método QIR pode melhorar consideravelmente um sistema de reconhecimento, até mesmo comparado com o RHE +GIC.

Figura 24: Resultados do método Estratégia Baseada em Região.
 FONTE: SHAN et al., (2003).

3.5 Método de Zhu et al. (2003)

Nesta seção é descrito o método para correção da iluminação de Zhu et al. (2003), que tem como objetivo obter uma melhor visualização de uma imagem, normalizando o seu contraste por meio de uma transformação afim, efetuando uma estimação do ganho e da iluminação do fundo.

Matematicamente, Zhu et al. (2003) modelaram mudanças de luz como uma transformação afim local do valor do *pixel*, de acordo com a equação (43), onde $G(x,y)$ é a imagem original, e $G'(x,y)$ é a imagem modificada como resultado da iluminação local.

$$G'(x, y) = A(x, y).G(x, y) + B(x, y) \quad (43)$$

Na equação (43), o efeito aditivo $B(x, y)$ é estimado com o filtro passa-baixa baseado em multi-resolução (seção 3.5.1), e o efeito multiplicativo $A(x, y)$ é tratado por meio do filtro homomórfico.

O algoritmo de normalização tenta eliminar os efeitos aditivos e multiplicativos causados na imagem resultante $G'(x, y)$, subtraindo-a do filtro passa-baixa estimado $\hat{B}(x, y)$. E, então, normalizando-a com a estimação do ganho $\hat{A}(x, y)$ (seção 3.5.2), de acordo com a equação (44), onde K é uma constante de valor não crítico.

$$\hat{G}(x, y) = \frac{k}{\hat{A}(x, y)}.(G'(x, y) - \hat{B}(x, y)) \quad (44)$$

Após essas operações é utilizado um métodos baseado em histograma para eliminar *pixels* que representam ruído.

A estimação da iluminação de fundo, a estimação do ganho e a normalização da imagem serão apresentadas nas seções seguintes.

3.5.1 Estimação da iluminação do fundo

O filtro passa-baixa para estimação da iluminação do fundo deveria ter uma máscara muito grande, o que teria um custo computacional considerável. Por esta razão Zhu et al. (2003) apresentaram uma aproximação do filtro passa-baixa baseado em multi-resolução, de menor custo computacional.

Uma estimação grosseira do filtro passa-baixa é efetuada, não sobrepondo o bloco da média da imagem, o que proporciona uma baixa resolução na representação da imagem. Uma re-escala simples desta imagem para a resolução original exibiria aspecto granulado (*blockiness*). Desta forma, Zhu et al. (2003) efetuaram a re-escala da imagem em estágios, onde cada resolução intermediária aproxima passo por passo a resolução mais próxima da original. A re-escala da imagem é feita através de passos, conforme ilustrado na Figura 25, criando uma interpolação *spline* multi-resolução da imagem reduzida.

Figura 25: Passos para re-escala da imagem.
FONTE: ZHU et al., (2003).

Uma vez que cada estágio dobra a largura e a altura da imagem de uma maneira específica ($2 * altura - 1$ e $2 * largura - 1$), é improvável que operações repetidas na imagem reduzida resultem numa imagem com as mesmas dimensões que a original. Como a imagem resultante excede o tamanho da original, ela deve ser redimensionada para o tamanho apropriado utilizando interpolação dos vizinhos mais próximos.

3.5.2 Estimação de ganho

Para a estimação do ganho reescreve-se a equação (43) para a seguinte forma:

$$G'(x, y) - B(x, y) = A(x, y).G(x, y) \quad (45)$$

O efeito multiplicativo conduz à filtragem homomórfica, e a estimação do ganho pode ser feita no domínio logarítmico, como ilustrado na equação (46).

$$\begin{aligned} f(x, y) &= \ln(G'(x, y) - B(x, y)) \\ &= \ln A(x, y) + \ln G(x, y) \end{aligned} \quad (46)$$

Com um filtro passa-baixa (LPF) aplicado em $f(x, y)$ será isolado $\ln A(x, y)$, podendo-se escrever:

$$LPF(f(x, y)) \approx \ln A(x, y) \quad (47)$$

Finalmente, a exponenciação é necessária para retornar ao domínio espacial normal.

$$A(x, y) = \exp(LPF(f(x, y))) \quad (48)$$

Para imagens de grandes dimensões essa implementação torna-se lenta.

3.5.3 Normalização da imagem

Para a normalização da imagem deve-se substituir os valores estimados das seções 3.5.1 e 3.5.2 na equação (44), com o passo adicional de re-escalar os dados para a faixa de valores permitidos [0,255].

Entretanto, observa-se que freqüentemente existem *pixels* ruidosos nos dois extremos do histograma, que não representam características importantes da imagem, mas impedem que a imagem utilize completamente a faixa dinâmica de visualização. Para evitar a redução do contraste, são estabelecidos dois limiares baseados na freqüência dos valores desses *pixels*. No histograma, o nível de cinza é pesquisado nos dois extremos até que o primeiro nível de cinza com alta freqüência seja encontrado. Os respectivos níveis de cinza são atribuídos como T_{lo} e T_{hi} , e matematicamente têm-se:

$$\hat{G}(x, y) = \begin{cases} g_{lo}, & \hat{G}(x, y) \leq T_{lo} \\ g_{hi}, & \hat{G}(x, y) \geq T_{hi} \\ \frac{g_{hi} - g_{lo}}{t_{hi} - t_{lo}} (\hat{G}(x, y) - T_{lo}) + g_{lo}, & \text{caso contrário} \end{cases} \quad (49)$$

Em alguns casos, nos quais a imagem original é ruidosa ou apresenta mudança abrupta de luz local, uma maneira fácil para compensar isto é modificar a estimacão do ganho para:

$$A'(x, y) = A(x, y) \cdot m(G'(x, y)) \quad (50)$$

onde $A(x, y)$ é o resultado da estimacão do ganho anterior e $m(.)$ é uma função de modificação para compensação que foi escolhida como $m(x) = \sqrt[4]{x}$.

A raiz quarta foi empregada com base em resultados experimentais. Como resultado, a imagem corrigida terá menos componentes de alta frequência.

Finalmente, a função de correção de iluminação pode ser escrita como:

$$\hat{G}(x, y) = N \left(\frac{k}{\hat{A} \sqrt[4]{G'(x, y)}} (G'(x, y) - \hat{B}(x, y)) \right) \quad (51)$$

onde $N(.)$ representa a normalização dos dados, com auxílio do método baseado em histograma descrito anteriormente.

Experimentos de Zhu et al. (2003) mostraram que o método de correção de iluminação pôde alcançar resultados satisfatórios na remoção de efeitos indesejáveis tais como sombra e variação de iluminação numa imagem (Figura 26). Porém, não foi capaz de obter bons resultados em faces com efeito acentuado de sombra.

Figura 26: Imagem original com diferentes iluminações na coluna 1, HE na segunda coluna e, resultado do método de correção de iluminação de Zhu et al. (2003) na terceira coluna.

FONTE: ZHU et al., (2003).

3.6 Normalização da iluminação utilizando a Transformadas de *Wavelets*

Du e Ward (2005) propuseram uma nova abordagem para normalização de iluminação em imagens de face utilizando a Transformada de *Wavelets*. A imagem resultante além de ter seu contraste melhorado também tem as bordas e detalhes realçados.

Nessa nova abordagem, a imagem é decomposta em altas e baixas frequências, e então é aplicada a equalização do histograma nos coeficientes de aproximação (baixa frequência), e ao mesmo tempo são acentuados os detalhes (alta frequência) multiplicando cada elemento da matriz dos coeficientes de detalhe (sub-bandas *LH*, *HL*, e *HH* da Figura 27) por um escalar maior do que um. Onde, *LH* são os coeficientes de detalhe horizontal, *HL* são os coeficientes de detalhe vertical, e *HH* são os coeficientes de detalhe diagonal. A imagem é reconstruída dos coeficientes de aproximação após a equalização do histograma e do alargamento dos coeficientes de detalhes, usando a Transformada *Wavelet* Inversa.

Figura 27: Estrutura de multi-resolução de decomposição *wavelet*.
FONTE: DU e WARD, (2005).

As imagens foram testadas na base de dados Yale (GEORGHIADES et al., 2000), e foi obtida uma melhora de 15,65% em relação ao método de equalização do histograma. Resultados dos experimentos estão ilustrados na Figura 28.

Com este método, um sistema de reconhecimento de faces é capaz de trabalhar com grandes variações na iluminação, levando em consideração o realce de contraste e borda. Além disso, facilita futuros passos de extração e reconhecimento, pois os coeficientes *wavelets* já foram obtidos.

Figura 28: Normalização com o método *wavelet*: (a) Imagens originais; (b) Imagens após a equalização do histograma. (c) Imagens resultantes.
FONTE: DU e WARD, (2005).

3.7 Compensação de iluminação por meio de Redes Neurais Artificiais (RNAs) e *wavelets*

Um método de compensação de iluminação em imagens de face baseado em Redes Neurais Artificiais (RNAs) e *wavelets* é proposto por Zhang et al. (2005).

Nesse método, inicialmente é realizada uma compensação de iluminação linear, para compensar as características de baixa frequência, baseada em filtragem homomórfica. Para refinar os resultados obtidos, é aplicado um método baseado em *wavelets* e redes neurais na imagem de face resultante. Essa técnica será descrita a seguir.

Considerando I_l a imagem obtida após a filtragem, sobre essa imagem é aplicada a Transformada de *Wavelet* para se obter o primeiro nível de decomposição, LLI_l , LHI_l , HLL_l e HHI_l (

Figura 29). Em seguida esses coeficientes são decompostos novamente para se obter o segundo nível. Os coeficientes LL , LH , HL ilustrados em vermelho na

Figura 29 são ajustados para compensar a iluminação da imagem utilizando RNAs.

Figura 29: Análise Wavelet.

FONTE: Adaptado ZHANG et al., (2005).

São utilizadas três redes separadamente, *rede 1* (cujos dados de treinamento são os coeficientes LL), *rede 2* (cujos dados de treinamento são os coeficientes LH) e *rede 3* (cujos dados de treinamento são os coeficientes HL). As três arquiteturas de rede são MLP (*Multi Layer Perceptron*) e têm o mesmo número de camadas (uma camada escondida e uma camada de saída).

Nesse trabalho, o tamanho da imagem utilizada foi 64×64 , a *wavelet* utilizada é a Biortogonal 3.3, o número de entradas é igual ao número de neurônios da camada de saída (196), e o número de neurônios da camada escondida é igual a 200.

No processo de treinamento foi escolhida a imagem de uma pessoa sob condições normais de iluminação, e foi aplicada a Transformada de *Wavelet*. Os coeficientes ilustrados em vermelho da

Figura 29 foram calculados, e estes foram denominados como padrão de LL (PLL), padrão de LH (PLH) e padrão de HL (PHL).

Os coeficientes LL , LH , HL das amostras, sob diferentes condições de iluminação, constituem o conjunto de entrada da *rede 1*, *rede 2* e *rede 3*, respectivamente. As saídas desejadas das três redes são, respectivamente, $(PLL - LL)$, $(PLH - LH)$, $(PHL - HL)$.

O resultado da compensação é: $(LL + \text{saída da rede 1})$, $(LH + \text{saída da rede 2})$, $(HL + \text{saída da rede 3})$. Após esses passos é aplicada a Transformada de *Wavelet* inversa.

O algoritmo de Zhang et al. (2005) pode ser sumarizado pelos seguintes passos:

Passo 1: Normaliza a imagem de face para a dimensão 64×64 .

Passo 2: Aplica o pré-processamento na imagem.

Passo 3: Aplica a Transformada de *Wavelet* Bior 3.3 para obter os coeficientes *LL*, *LH* e *HL*, conforme está ilustrado em vermelho na

Figura 29.

Passo 4: Aplica nas entradas das redes 1, 2 e 3, os coeficientes *LL*, *LH* e *HL*, respectivamente, do passo anterior.

Passo 5: Substitua *LL* por $(LL + \text{saída da rede 1})$, *LH* por $(LH + \text{saída da rede 2})$, *HL* por $(HL + \text{saída da rede 3})$.

Passo 6: Aplique a Transformada *Wavelet* Inversa para obter a imagem normalizada.

O algoritmo desenvolvido por Zhang et al. (2005) melhora o efeito de iluminação tanto na detecção quanto no reconhecimento de faces sobre diferentes tipos iluminação. Além disso, o método é bastante robusto e pode se usado em várias aplicações.

Figura 30: Alguns resultados do algoritmo que utiliza RNA utilizado para a compensação da iluminação.

FONTE: ZHANG et al., (2005).

3.8 Variação do filtro homomórfico

Delac et al. (2006) propuseram uma adaptação no filtro homomórfico (maiores detalhes sobre este tipo de filtro encontram-se na seção 2.2.2) para realçar a reflectância numa dada imagem de face, enquanto reduz a contribuição da iluminação, com a finalidade de melhorar o desempenho de sistemas de reconhecimento de faces.

A imagem original foi dividida em sub-imagens, e filtros foram aplicados em cada sub-imagem individualmente (Delac et al., 2006). A imagem foi inicialmente dividida verticalmente em duas metades (obtendo-se duas sub-imagens da imagem original), e aplicado o filtro em cada metade. Seguindo esta idéia, a imagem original foi dividida na horizontal e novamente foi aplicado o filtro em cada metade (Figura 31).

Figura 31: Variação do filtro homomórfico.
FONTE: Adaptado DELAC et al., (2006).

Seja $I_{HMV}(x,y)$ o resultado da imagem dividida verticalmente, $I_{HMH}(x,y)$ o resultado da imagem dividida horizontalmente, e $I_{HMMOD}(x,y)$ a imagem resultante da variação do filtro homomórfico. Pode-se então escrever que:

$$I_{HMMOD}(x,y) = \frac{1}{2} \cdot [I_{HMV}(x,y) + 0.75 \cdot I_{HMH}(x,y)] \quad (52)$$

Desde que I_{HMV} obteve melhores resultados do que I_{HMH} nos testes, Delac et al. (2003) decidiram manter I_{HMV} e multiplicar I_{HMH} por uma constante igual a 0.75 (escolhido baseado nos resultados dos experimentos), para diminuir sua influência na representação final. Esta combinação produziu resultados satisfatórios nos experimentos e foi mantido como representação final.

Para comprovar a eficiência do método, este foi comparado com outros métodos. A Figura 32 ilustra os resultados dessa comparação. O filtro homomórfico com realce de contraste consistiu da aplicação do filtro homomórfico seguido de uma versão modificada da Equalização do Histograma (HE).

Foi comprovado, nos experimentos de Delac et al. (2006), que o método apresentado gera bons resultados (melhora de 13,4% no reconhecimento, comparado com o método de

equalização do histograma), e justifica futuras pesquisas para utilização do filtro homomórfico para tarefas de melhoria da iluminação em imagens.

Figura 32: Realce de imagem: (a) Imagem original, (b) Utilizando HE, (c) Utilizando filtro homomórfico, (d) Utilizando filtro homomórfico e realce de contraste, (e) Utilizando filtro homomórfico das duas sub-imagens verticais separadamente, (f) Utilizando filtro homomórfico das duas sub-imagens horizontais separadamente, (g) Utilizando a variação do filtro homomórfico.

FONTE: DELAC et al., (2006).

3.9 Retinex

A teoria do retinex desenvolvida por Land e MacCann (1971) descreve como o sistema visual humano percebe a cor/tonalidade da cena natural. Em seus experimentos, foi percebido que a sensação de cor tem relação arbitrária com a composição de iluminação em termos do comprimento de onda e luminância, e que a sensação de cor é dependente da reflectância da superfície. Portanto, o objetivo da teoria do retinex é entender e encontrar um modelo computacional de como a visão processa o estímulo físico, de tal maneira que a consistência de cor seja preservada. Para este propósito, duas suposições foram estabelecidas (HEUSCH et al., 2005):

1- O sistema de visão realiza a mesmo cálculo, independentemente, em cada canal de cor.

2 – Em cada canal, a intensidade do sinal é proporcional ao produto da iluminação pela reflectância da superfície:

$$I(x, y) = L(x, y)R(x, y) \quad (53)$$

Existem diferentes versões do retinex, que foram melhoradas progressivamente, e dentre elas pode-se citar o Retinex de Única Escala (SSR), o Retinex de Múltipla Escala (MSR), e o Retinex de Múltipla Escala com Restauração de Cor (MSRCR). Nas subseções a

seguir, serão mencionados trabalhos relacionados a essas versões, juntamente com o embasamento teórico.

3.9.1 Retinex de Única Escala (SSR)

Jobson et al. (1997) definiram uma implementação prática do retinex, o Retinex de Única Escala (SSR) e encontraram a melhor versão do *gain/offset*, que é aplicado após a operação do retinex.

O SSR é representado pela seguinte equação:

$$R_i(x, y) = \log I_i(x, y) - \log[F(x, y) * I_i(x, y)] \quad (54)$$

onde $i \in R, G, B$, I é a imagem de entrada, F é a função *surround*, e $R_i(x, y)$ é a imagem de saída. Após calcular $R_i(x, y)$, é aplicada a correção *gain/offset*, para que a imagem fique com brilho adequado.

A correção *gain/offset* é um método de realce de imagem para expandir a faixa dinâmica da imagem, de maneira a adequá-la à faixa dinâmica do meio de visualização.

Para uma cena com faixa dinâmica entre $r_{\max}$ e $r_{\min}$, este método pode ser expresso como a seguir:

$$I'_i(x, y) = \frac{d_{\max}}{r_{\max} - r_{\min}} \cdot (I_i(x, y) - r_{\min}) \quad (55)$$

onde $d_{\max}$ é a faixa dinâmica de exibição, normalmente 255, I_i é a i -ésima entrada, e I'_i é a i -ésima saída.

Jobson et al. (2007) avaliaram modelos da função *surround* do retinex, e a função Gaussiana, apresentada na equação (56), foi a que apresentou melhor resultado, quando comparada com outras funções.

$$F(x, y) = K \cdot \exp(-r^2 / c^2) \quad (56)$$

Na função *surround* da equação (56), $r = \sqrt{x^2 + y^2}$, c é a constante do espaço Gaussiano da função, e K é um valor selecionado de forma que:

$$\iint F(x, y) dx dy = 1 \quad (57)$$

A Figura 33 ilustra os resultados obtidos com três funções *surround* diferentes, podendo-se observar que a Gaussiana é a que resultou na melhor imagem.

Figura 33: Resultados da aplicação do SSR utilizando três funções *surround*: inverso do quadrado, exponencial e gaussiana.
FONTE: JOBSON et al., (1997).

3.9.2 Retinex de Múltipla Escala (MSR)

No SSR existe um compromisso controlado pela escala da função *surround* entre a compressão da faixa dinâmica e a representação da tonalidade, e ao se melhorar uma a outra é reduzida. É possível obter imagens com melhor cor/tonalidade ou compressão da faixa dinâmica, mas não os dois simultaneamente.

O Retinex de Múltipla Escala (MSR) supera essa limitação para a maioria das cenas, combinando várias saídas SSR para produzir uma única imagem de saída, resultando em boa compressão da faixa dinâmica e constância de cores, além de boa representação de tonalidade (RAHMAN et al., 1996).

O MSR pode ser descrito da seguinte forma:

$$R_{MSR_i}(x, y) = \sum_{n=1}^N W_n \cdot \{\log[I_i(x, y)] - \log[I_i(x, y) * M_n(x, y)]\} \quad (58)$$

onde $i \in R, G, B$ representa as três bandas de cores, N é o número de escala usada, e W_n são os fatores de pesos para as escalas. $M_n(x, y)$ é a função *surround*, definida como:

$$M_n(x, y) = K_n \cdot \exp(-(x^2 + y^2) / \sigma_n^2) \quad (59)$$

onde σ_n é o desvio padrão da distribuição Gaussiana que determina a escala.

A Figura 34 ilustra uma comparação feita por Rahman et al. (1996), entre o SSR e o MSR. Na imagem gerada pelo MSR percebe-se detalhes que não são evidentes na imagem gerada pelo SSR.

Figura 34: Realce de imagem com o algoritmo retinex: a) Imagem original b) Utilizando o método SSR c) Utilizando o método MSR.

FONTE: RAHMAN et al., (1996).

3.9.3 Retinex de Múltipla Escala com Restauração de Cor (MSRCR)

O Retinex de Múltiplas Escalas com Restauração de Cor (MSRCR) é uma versão do MSR que combina compressão de faixa dinâmica e constância de cor, com um filtro de “restauração” de cor, que proporciona uma boa execução de cor. Este método foi testado com um número grande de imagens no trabalho de Rahman (1997), onde foi comprovado,

consistentemente, que o MSRCR possuir melhor qualidade das imagens do que qualquer método de realce de imagem convencional.

O MSRCR pode ser descrito da seguinte forma:

$$R_{MSRCR_i}(x, y) = \sum_{n=1}^N W_n \cdot \{\log[I_i(x, y)] - \log[I_i(x, y) * M_n(x, y)]\} \quad (60)$$

onde $M_n(x, y)$ é a função *surround*, definida como (RAHMAN et al., 1996):

$$M_n(x, y) = K_n \cdot \exp(\sigma_n^2 / (x^2 + y^2)) \quad (61)$$

O MSRCR foi comparado com métodos que são amplamente utilizados para o realce de imagem, tais como correção gama, aplicações *gain/offset*, métodos de modificação do histograma, dentre outros. Concluiu-se que, para o conjunto de imagens por eles utilizadas, parte destes métodos mostraram-se satisfatórios para algumas imagens, mas somente o MSRCR apresentou bons resultados para todas as imagens do conjunto. A Figura 35 ilustra os resultados dos testes.

Figura 35: Realce de imagem com o MSRCR: a) Imagem original. b) Imagem após equalização do histograma. c) Resultado.

FONTE: RHAMAN et al., (1997).

3.10 Considerações finais

Neste capítulo foram apresentados trabalhos relacionados à compensação de iluminação para imagens de face. Apesar dos inúmeros trabalhos já publicados, os problemas relacionados à compensação de iluminação ainda não foram resolvidos completamente, de

forma satisfatória. Os artigos apresentados neste capítulo evidenciam que, métodos mais recentes como *wavelets* e redes neurais artificiais têm sido estudadas para serem aplicadas na solução dos problemas relacionados à iluminação.

O presente trabalho baseou-se principalmente nos seguintes métodos apresentados nesse capítulo: LogAbout (LIU et al., 2002), filtro homomórfico modificado (DELAC et al., 2006) e o método baseado em *wavelets* (DU e WARD, 2005).

4 Reconhecimento de Faces

4.1 Considerações iniciais

Neste capítulo são apresentados os seguintes métodos de reconhecimento facial utilizados neste trabalho: PCA, autôfaces e *wavelets* com Redes Neurais Artificiais.

4.2 Métodos para o reconhecimento facial

Os métodos para reconhecimento de faces são agrupados nas seguintes categorias (ZHAO, W. et al. 2003):

- métodos Holísticos: usam toda região da face como dado de entrada para sistemas de reconhecimento;
- métodos baseados em características locais: primeiramente extrai características locais tais como olhos, nariz e boca, e em seguida suas posições e estatísticas locais (geométricas e/ou aparência) são alimentados em um classificador estrutural;
- métodos híbridos: usa características locais e toda região da face para tarefas de reconhecimento facial.

Em Brunelli e Poggion (1993) foi realizado um estudo comparativo entre o método *template matching* e o método por atributos locais. O método holístico obteve melhores resultados do que os locais. Uma das vantagens de se utilizar abordagens globais é o fato de pequenas variações locais não prejudicarem o reconhecimento. A iluminação, e em alguns casos o custo computacional, são as principais desvantagens de tais métodos.

Dentre os métodos holísticos apresentados em Zhao (2003) encontram-se os métodos baseados em Análise de Componentes Principais (PCA), Análise Discriminante Linear (LDA), Análise de Componente Independente (ICA), Máquinas de Vetores Suporte (SVM), etc. Esses métodos baseiam-se em características estatísticas, tendo sua própria forma de representação no espaço para formar um espaço dimensional menor baseado em conceitos estatísticos diferentes. Obtêm-se os coeficientes com a nova representação de características de cada imagem de face, projetando-se o vetor de faces nos vetores da base do sistema. A partir de um classificador, analisam-se os coeficientes dos protótipos das faces de treinamento com os do teste e, de acordo com um limiar estabelecido, é definido se a face está ou não no banco (GASPAR, 2006).

Os fatores que influenciam o desempenho dos métodos híbridos incluem, determinar quais características devem ser combinadas e como combiná-las (TAN et al., 2006).

Em Pentland et al. (1994), é explorado o uso de método híbrido combinando auto-faces e extração de características, tais como: olhos, boca, e nariz. Os experimentos mostraram alguma melhoria com esse método.

Em Tan et al. (2006) é apresentada uma revisão sobre métodos para reconhecimento de faces. Nesse artigo é mencionado que os métodos baseados em características locais para o reconhecimento de faces, geralmente utilizam quatro passos: partição de regiões locais, extração de característica local, seleção de característica e classificação. Estes passos estão ilustrados na Figura 36.

Figura 36: Visão geral de métodos baseados em aparência local.
FONTE: TAN et al., (2006).

A partição de regiões locais envolve, por exemplo, a forma e o tamanho das regiões locais. A maneira mais simples e largamente utilizada é a janela retangular, que pode ou não ser sobreposta. O tamanho das áreas locais tem influência direta no número de características locais.

A extração de características locais determina como representar as regiões locais definidas. As características freqüentemente utilizadas incluem: os valores dos níveis de

cinza, características obtidas com *wavelets* de Gabor, com *wavelet* de Haar, com fractais, e com outros métodos.

Se muitas das características forem geradas no passo de extração de característica, será necessário um estágio adicional de seleção de características. A Análise de Componentes Principais (PCA) é um método geralmente utilizado, garantindo perda mínima de informação. Pode também ser utilizada a Análise de Discriminante linear (LDA) e alguma estatística local.

O passo final é a identificação da face. Combinar classificadores é a maneira mais comum para este propósito. Em particular, cada componente classificador é aplicado sobre uma característica local, e então a decisão final é feita pela votação majoritária ou pela soma ponderada linear.

4.3 Reconhecimento de faces utilizando PCA (Análise de Componentes Principais)

PCA é um método para reconhecimento de padrões numa determinada base de dados, e expressa as similaridades e diferenças dessas imagens a respeito de uma imagem alvo. Uma imagem 2-D, descrita por uma determinada matriz, pode ser representada por um vetor $X \in R$, caracterizando as imagens num espaço de menor dimensão, onde apenas as informações mais relevantes para o reconhecimento estarão presentes.

Sabendo-se que cada imagem de face é definida por uma matriz de *pixels* com intensidades na escala de cinza, e definido os vetores $T = (T_1, T_2, \dots, T_M)$ associados aos *pixels* das imagens concatenados por coluna, a média Ψ é calculada da seguinte forma (SANTOS, 2005):

$$\Psi = 1/M \sum_{n=1}^M T_n \quad (62)$$

Através da equação (62), são calculados os desvios de cada imagem em relação a média:

$$\Phi_n = T_n - \Psi \quad (63)$$

O conjunto de vetores definidos por Φ está normalizado em relação a média, e pronto para ser utilizado no cálculo dos auto-vetores μ_k e auto-valores λ_n da matriz de covariância da equação (64).

$$C = 1/M \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T \quad (64)$$

onde a matriz $A = \{\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_n\}$

Considere V_n os auto-vetores de $A^T.A$ semelhantes a:

$$A^T.A.V_n = \lambda_n.V_n \quad (65)$$

com a multiplicação de ambos os lados por A , obtém-se:

$$A.A^T.A.V_n = \lambda_n.A.V_n \quad (66)$$

concluindo-se que $A.V_n$ são os auto-vetores de $C = A.A^T$.

Seguindo esta análise, a matriz $L = A^T.A$ é construída, resultando numa matriz de dimensões $M \times M$, e $L_{m,n} = \phi \phi_n^T$. Encontram-se os M auto-vetores V_n de L . Estes auto-vetores determinam uma combinação linear das M imagens do conjunto de treinamento para formar os auto-vetores μ_n :

$$\mu_n = \sum_{k=1}^M V_{nk} \phi_k = A.V_n \quad n=1, 2, \dots, M \quad (67)$$

A partir desta análise, os cálculos são reduzidos da ordem dos números de *pixels* nas imagens (N^2) para o número de imagens no conjunto de treinamento (M). Na prática é usado um número pequeno de faces para o treinamento ($M < N^2$), tornando o cálculo mais gerenciável. Com os auto-vetores associados é possível colocar em ordem estes, de acordo com sua utilidade na caracterização da variação entre as imagens, uma vez que os primeiros

auto-vetores possuem valores altos, caracterizando a sua representatividade com relação aos dados (SANTOS, 2005).

4.3.1 Algoritmo de reconhecimento utilizado

Os passos do algoritmo de reconhecimento facial aplicando PCA para redução de dimensionalidade, e utilizado nesse trabalho, são descritos na Tabela 2.

Tabela 2: Algoritmo de reconhecimento utilizando PCA e rede neural LVQ
Algoritmo de reconhecimento utilizando PCA e LVQ

- 1 – As imagens são armazenadas num vetor onde cada coluna representa uma imagem.
- 2 – Cálculo da média total de todas as imagens utilizadas no treinamento.
- 3 – A matriz de covariância é calculada.
- 4 – Com a matriz de covariância é possível encontrar os auto-vetores e auto-valores.
- 5 – As médias das imagens de cada classe são então calculadas e projetadas no espaço PCA, compondo assim o conjunto de dados de entrada para o treinamento da rede neural.
- 6 – Os dados de 5 são as entradas da rede neural LVQ, a saída são as respectivas classes da imagem, por exemplo, classe 1, 2, 3...

FONTE: FERNANDES e GONZAGA, (2007).

4.4 Algoritmo de reconhecimento utilizando auto-faces

O método de auto-faces é bastante parecido com o método PCA, ou seja, é calculada a média das imagens, só que aqui cada linha é uma face, a imagem média é subtraída de cada imagem (Φ), os auto-vetores (u) e auto-valores são calculados baseando-se na matriz de covariância. Porém, neste método é criada a seguinte matriz de transformação:

$$\mu = u\Phi \quad (68)$$

onde a matriz μ contém $M-I$ auto-vetores significativos. É possível eliminar alguns $M-I$ auto-vetores pela ordem de importância dos seus correspondentes auto-valores, gerando M' auto-vetores (PENHARBEL et al., 2005). E para finalizar é preciso normalizar os M' vetores da matriz μ .

No procedimento de classificação, a face é transformada em componentes de auto-vetores. Primeiramente é comparada a imagem de entrada com a imagem média, e sua diferença é multiplicada por cada auto-vetor da matriz L . Cada valor representará um peso e é salvo no vetor Ω (SERRANO, 2007).

$$\omega_k = u_k (\Gamma - \Psi) \quad \Omega^T = [\omega_1, \omega_2, \dots, \omega_n] \quad (69)$$

A determinação de qual classe proporciona a melhor descrição para a imagem de entrada é feita pela distância Euclidiana mínima:

$$\varepsilon_k = ||\Omega - \Omega_k||^2 \quad (70)$$

A face de entrada é considerada pertencente ao banco, se ε_k estiver abaixo de um determinado limiar.

No algoritmo de reconhecimento facial utilizado neste trabalho, para encontrar o limiar das imagens de treinamento para uma determinada face da base de dados, foi preciso gerar um vetor com todas as distâncias mínimas de cada imagem de treinamento e depois foi identificado a média desse vetor.

4.5 Reconhecimento facial utilizando rede neural MLP e Wavelets

No trabalho de Rizk et al. (2004) é utilizada a Transformada Discreta de *Wavelet* para extração dos dados e para redução da dimensionalidade. Em seguida é utilizada uma rede MLP que é treinada com elementos horizontais da imagem, e uma Rede Neural de Base Radial (RBF) que é treinada com os elementos verticais da imagem. As duas redes foram treinadas com o mesmo conjunto de faces, tendo sido utilizado um total de 200 faces.

No processo de reconhecimento facial, cada rede gera 40 saídas. Foi elaborado um algoritmo que seleciona as cinco saídas de cada rede com os valores de máximo. Em seguida o algoritmo compara os valores de máximo de cada rede até encontrar dois valores idênticos, e este corresponde à face encontrada.

Neste trabalho foi utilizada a base de dados ORL, que contém dez diferentes imagens, em tons de cinza, com resolução de 92×112 pixels, de 40 faces distintas. Nesta base, há variações na expressão facial (olhos fechados/abertos, sorindo/não sorindo), e detalhes faciais (com/sem óculos).

Foi utilizada a Transformada Discreta de *Wavelet* com três níveis de decomposição, reduzindo o espaço de entrada de 10304 (92×112) para 1024 (32×32) unidades. O espaço de entrada foi normalizado, dividindo-se os dados pela raiz quadrada da soma do quadrado dos coeficientes.

As características da rede MLP utilizada foram as seguintes: 1024 entradas, uma camada escondida com 50 neurônios e 40 neurônios na camada de saída. Essa rede foi treinada com cinco amostras de cada pessoa, e as demais amostras foram utilizadas para teste. Para a rede RBF foram utilizadas duas camadas, 1024 entradas, tendo sido usada 175 épocas para alcançar seu melhor desempenho. O seu treinamento foi feito de forma similar à rede MLP.

A rede MLP atingiu 92% de acerto no reconhecimento, enquanto que a RBF obteve 86.75%. Quando não foi utilizada a Transformada de *Wavelet*, a rede MLP obteve uma taxa de 85.75%, enquanto que a taxa da RBF foi de 83.5%.

4.6 Considerações finais

Neste capítulo foi apresentada uma visão geral sobre os métodos holísticos, os baseados em características e os métodos híbridos utilizados para o reconhecimento de faces.

Os métodos holísticos são os que apresentam melhores resultados no reconhecimento. Por outro lado, são sensíveis às variações de posição e escala.

Em seguida foram apresentados os algoritmos de reconhecimento facial aplicados neste trabalho, que se enquadram principalmente nos métodos holísticos.

5 Metodologia

5.1 Considerações iniciais

Este capítulo apresenta a metodologia utilizada para a comparação entre os seguintes métodos de compensação da iluminação em imagens de face: LogAbout (LIU et al., 2002), variação do filtro homomórfico (DELAC et al., 2006) e método baseado em *wavelets* (DU e WARD, 2005).

5.2 Banco de imagens

Foi utilizado o banco *Yale Face Database B* (GEORGHIADES et al., 2000), constituído por 5760 imagens de 10 pessoas distintas e com diferentes posições da fonte de luz. Há 9 poses diferentes para cada pessoa sendo que cada pose possui 64 diferentes condições de iluminação. A iluminação varia com relação à direção da fonte de luz (grau azimute) e graus de elevação da iluminação. As imagens desse banco encontram-se em tons de cinza (256), e possuem tamanho de 640x480 *pixels*.

As imagens utilizadas desse banco foram previamente recortadas para se obter a região de interesse (nariz, boca e olho), como pode ser observado na Figura 37, resultando em imagens com dimensões de 250x250 *pixels*. Posteriormente, essas imagens foram redimensionadas para 256x256, pois um dos algoritmos de normalização (variação do filtro homomórfico) requer imagens de entrada na ordem de $2^n \times 2^n$.

Figura 37: Imagem original do banco Yale Face Database B em (a), e imagem do banco Yale Face Database B após recorte da região de interesse em (b).

Foram escolhidas as imagens frontais do banco Yale e estas foram divididas em três subconjuntos com 21 imagens de cada pessoa, totalizando 210 imagens em cada subconjunto. Esta divisão foi efetuada com a finalidade de diminuir o número de imagens por pessoa, e assim, agilizar o processamento e facilitar o trabalho dos algoritmos de reconhecimento de faces. As imagens do subconjunto 1 são as imagens de face onde o grau de elevação da iluminação está entre -40 graus e -10 graus; as imagens do subconjunto 2 possuem grau de elevação entre 0 graus e 15 graus; as do subconjunto 3 estão entre 20 graus e 90 graus. Na Figura 38 encontra-se ilustrado uma pessoa de cada subconjunto, após selecionada a região de interesse da face.

Figura 38: Imagens de face de uma pessoa de cada subconjunto. a) Subconjunto 1. b) Subconjunto 2. c) Subconjunto 3.

5.3 Metodologia

No desenvolvimento deste trabalho foi utilizado o MATLAB (*Matrix Laboratory*) na versão 7.0, com os respectivos *Toolbox* de processamento de imagem, *wavelets* e redes neurais. O ambiente de programação MATLAB permite o uso de sua própria linguagem de programação, com chamadas de funções de processamento de imagem, agilizando o desenvolvimento dos algoritmos, sendo por esse motivo utilizado.

O trabalho desenvolvido consistiu na implementação de três algoritmos de normalização da iluminação, que foram avaliados com três algoritmos de reconhecimento facial. Um diagrama esquematizando o processo de desenvolvimento pode ser encontrado na Figura 39.

Após as imagens do banco terem sido recortadas na região de interesse e redimensionadas para 256×256 , conforme mencionado na seção 5.2, foi aplicado sobre estas, conforme mostrado na Figura 38, cada algoritmo de normalização de iluminação. As imagens resultantes foram utilizadas como entrada nos seguintes algoritmos de reconhecimento de

faces: auto-faces (SERRANO, 2007), PCA com rede neural LVQ (FERNANDES e GONZAGA, 2007) e rede neural com *wavelets* (RIZK et al., 2004).

Figura 39: Diagrama com os algoritmos de normalização de iluminação e reconhecimento facial utilizados no trabalho.

Nas próximas subseções será descrita a metodologia de cada etapa desenvolvida.

5.3.1 Filtro LogAbout

O primeiro método utilizado para a normalização da iluminação foi o LogAbout (LIU et al., 2002), descrito na seção 3.3, que consistiu, primeiramente, da aplicação de um filtro *high-boost* na imagem original (Figura 40), seguida da seguinte transformação logarítmica:

$$g = c.\log(a + 1) \quad (71)$$

onde c é uma constante cujo valor é 8,3901, determinado por Liu et al. (2002), a é a imagem de face e g é o resultado da aplicação da transformação logarítmica.

-1	-1	-1
-1	9	-1
-1	-1	-1

Figura 40: Máscara utilizada.
FONTE: LIU et al., (2002).

No resultado da aplicação logarítmica é efetuado o alargamento de contraste, obtendo-se assim o resultado final, a imagem normalizada. A Figura 41 ilustra o resultado de imagens de face em cada etapa deste método.

Figura 41: Normalização com o método LogAbout: a) Imagens originais. b) Imagens após transformação logarítmica. c) Imagens finais com alargamento de contraste, ou seja, imagens resultantes.

Além do método LogAbout realçar as imagens, este método obteve bons resultados quando aplicado em imagem escura ou clara, como pode ser observado na Figura 42 (a), (b) e (c). Porém, o resultado não foi satisfatório para a face da Figura 42 (d), que contém efeito horizontal de luz e sombra, o qual foi ressaltado no resultado.

Figura 42: Imagens originais na primeira linha, e imagens após serem normalizadas com o método LogAbout na segunda linha.

5.3.2 Transformada de *Wavelet*

O segundo método utilizado foi o da Transformada de *Wavelet*, que se baseou no trabalho de Du e Ward (2005), apresentado na seção 3.6. Consistiu em aplicar a Transformada de *Wavelet* de Daubechies (Daub 10), nas imagens do banco de faces utilizado, com a finalidade de normalizar a iluminação de cada imagem.

Na aplicação da Daub 10, cada imagem sofre dois níveis de decomposição, conforme pode ser observado na Figura 43.

Figura 43: Decomposição *wavelet* de uma imagem: a) Imagem original. b) Primeiro nível de decomposição. c) Segundo nível de decomposição.

Em seguida é aplicada a equalização de histograma no coeficiente de aproximação LL2, e os coeficientes de detalhes (HL2, LH2, HH2, HL1, LH1, HH1), ilustrados na Figura 44, são multiplicados por um fator de escala. De acordo com Du e Ward (2005), esse fator de escala foi determinado experimentalmente, podendo assumir valores entre 80 e 100.

Para avaliação do resultado dessa normalização com o algoritmo PCA de reconhecimento de faces foi utilizado um fator de escala igual a 90, uma vez que nos experimentos de Du e Ward (2005) valores dessa ordem obtiveram melhores resultados, e nos experimentos efetuados isto foi comprovado. No entanto, quando se avaliou o resultado dessa normalização com o algoritmo MLP de reconhecimento de faces, e o de auto-faces, foi utilizado o valor 3, pelo fato de valores acima deste resultarem em baixas taxas de reconhecimento.

Figura 44: Resultado final da decomposição *wavelet*.

A imagem normalizada resultante é reconstruída a partir do coeficiente de aproximação (LL2) e dos coeficientes de detalhes em todas as três direções, usando a Transformada *Wavelet* Inversa. Um exemplo do resultado encontra-se na Figura 45.

Figura 45: Normalização com o método *wavelet*: a) Imagens originais. b) Resultados.

O método *wavelet* realça as bordas e o contraste de uma imagem, além de obter bons resultados em imagens escuras e claras, porém tende a escurecer um pouco a imagem, como pode ser observado na Figura 46 (e).

Figura 46: Imagens originais em (a) e (d), imagens após método *wavelet* com fator de escala igual a 3 em (b) e (e), e imagens após método *wavelet* com fator de escala igual a 90 em (c) e (f).

5.3.3 Filtro homomórfico

O terceiro método utilizado para a normalização de iluminação foi a variação do filtro homomórfico, apresentada no trabalho de Delac et al. (2006), descrito na seção 3.8.

Para a aplicação desse filtro, a imagem foi inicialmente dividida verticalmente, em duas partes, e o filtro foi aplicado em cada parte. Em seguida, a imagem original foi dividida horizontalmente em duas partes, aplicando-se o filtro em cada parte.

A imagem resultante foi obtida pela união das sub-imagens verticais e horizontais, conforme mencionado na seção 3.8. Neste método foi utilizado o filtro Butterworth com frequência de corte $D_o = 0.25$ e $n = 1$ (DELAC et al., 2006).

Na Figura 47 é apresentado o resultado da normalização utilizando esse método.

Figura 47: Imagens originais na primeira linha, e imagens após serem normalizadas com o método variação do filtro homomórfico na segunda linha.

Apesar do método de normalização variação do filtro homomórfico apresentar uma melhoria na iluminação para boa parte das imagens, como exemplificado pelas Figuras 47 (a) e (b), este não apresentou resultados satisfatórios em imagens muito escuras, como pode ser observado na Figura 47 (c).

5.3.4 Tempo dos métodos de compensação da iluminação

A Tabela 3 apresenta o tempo de processamento médio de uma imagem, que foi determinado a partir do tempo de execução das 210 imagens do subconjunto 1 dividido por 210, para cada método de normalização da iluminação, executado num computador AMD Athlon XP, de 2.2 GHz e com 1 GB de Memória RAM. O menor tempo de normalização obtido foi com o método LogAbout, seguido de *wavelet*. O maior tempo de normalização da iluminação foi para variação do filtro homomórfico.

Tabela 3: Tempo de processamento dos métodos de normalização de iluminação utilizados

Método	Tempo (em segundos)
LogAbout	0,0717
Variação do filtro Homomórfico	3,0248
<i>Wavelet</i>	0,6225

6 Resultado e discussões

6.1 Considerações iniciais

Neste capítulo são apresentados os resultados obtidos com a aplicação de cada método de normalização, e a avaliação desses métodos através dos seguintes algoritmos de reconhecimento de faces: auto-faces (SERRANO, 2007), PCA com rede neural LVQ (FERNANDE e GONZAGA, 2007) e rede neural com *wavelets* (RIZK et al., 2004).

Finalmente, é apresentada uma discussão sobre a taxa de acerto de cada algoritmo de reconhecimento de faces aplicado sobre as imagens do banco de dados.

6.2 Avaliação dos resultados através dos algoritmos de reconhecimento de faces

Com a finalidade de efetuar uma avaliação de qual método de normalização de iluminação é mais eficaz para o reconhecimento facial, foram utilizados os seguintes algoritmos de reconhecimento facial: auto-faces (SERRANO, 2007), PCA com rede neural LVQ (FERNANDES E GONZAGA, 2007) e *wavelets* com rede neural MLP (RIZK et al., 2004).

Cada método de normalização da iluminação foi testado com cada algoritmo de reconhecimento, separadamente.

Em todos os algoritmos foram utilizadas 16 imagens de cada pessoa (totalizando 160 imagens) para o treinamento e 5 para teste (totalizando 50 imagens), em cada subconjunto.

Para melhor avaliação dos resultados foram geradas as matrizes de confusão para cada algoritmo de reconhecimento.

6.2.1 Algoritmo utilizando PCA e rede neural LVQ

No algoritmo de reconhecimento facial que utiliza PCA e rede neural LVQ (FERNANDES e GONZAGA, 2007), a dimensão do espaço PCA utilizado foi igual a 60 características relevantes. Esse algoritmo encontra-se descrito na seção 4.3.

O gráfico da Figura 48 ilustra a taxa de acerto do algoritmo de reconhecimento utilizando PCA e rede neural LVQ nos subconjuntos 1, 2 e 3. A Tabela 4 mostra a taxa de melhora do algoritmo de reconhecimento para cada método de normalização da iluminação, comparando-se com a não utilização da normalização. O símbolo “-” da tabela indica que não ocorreu melhora nas taxa de reconhecimento. Pelos resultados apresentados, fica evidente que o método que utiliza *wavelet* apresentou melhores resultados, ressaltando-se a diferença considerável dos resultados, quando comparado com os outros métodos. Além disso, avaliando-se se foi vantajoso ou não o uso de um método de normalização para o reconhecimento facial, conclui-se que houve uma melhoria considerável na taxa de reconhecimento quando se usa o método de *wavelets* e o método LogAbout. Por outro lado, o filtro homomórfico resultou em melhoria insignificante na taxa de reconhecimento.

Figura 48: Gráfico do método que utiliza PCA e LVQ.

Tabela 4: Resultado do método que utiliza PCA e LVQ.

Métodos aplicados	% de melhora em relação à imagem original			
	Subconjunto 1	Subconjunto 2	Subconjunto 3	Média
LogAbout	32	22	12	22
Variação do filtro homomórfico	2	-	4	2,3
<i>Wavelet</i>	46	30	58	44,6

O algoritmo PCA tem como finalidade a redução de dimensionalidade, extraindo apenas as características mais significantes de uma dada imagem. Como o método de normalização *wavelet* realça as bordas e o contraste de uma imagem, a utilização do algoritmo PCA com LVQ obteve os melhores resultados com este algoritmo. O filtro homomórfico também realça as características das imagens, porém, em imagens escuras não obteve bons resultados (Figura 47 (c)), e este fato provavelmente influenciou para que ele obtivesse taxa de acerto inferior aos demais. O método LogAbout conseguiu melhorar tanto as imagens escuras quanto as claras (Figura 42), obtendo por este motivo o segundo melhor resultado no algoritmo de reconhecimento.

Será considerada a seguir a avaliação através da matriz de confusão para o algoritmo de reconhecimento facial PCA com LVQ (anexo A, página 89), gerada para cada método de normalização da iluminação e gerada também para as imagens sem correção de iluminação, aplicados sobre os subconjuntos 1, 2 e 3. Na matriz de confusão foram consideradas as 10 pessoas utilizadas para teste, com cinco imagens para cada pessoa.

Para o subconjunto 1 pode-se concluir que, sem a utilização de um método de correção de iluminação, o algoritmo identifica erroneamente, de forma freqüente, várias pessoas como sendo a pessoa 6, e isso ocorre de forma menos freqüente, com a pessoa 4. Com o método variação do filtro homomórfico, o algoritmo identifica erroneamente, de forma freqüente, as outras pessoas com sendo a pessoa 10, e isso ocorre de forma menos freqüente, com a pessoa 4. Ocorre também uma identificação incorreta da pessoa 3 como sendo a pessoa 2. Com o método LogAbout, o algoritmo identifica erroneamente, de forma menos freqüente, as outras pessoas como sendo a pessoa 10, classifica erroneamente, com menos freqüência, as outras pessoas como sendo a pessoa 6 e 9, e classifica, apenas uma vez, outras pessoas como sendo a

pessoa 4 e 5. Finalmente, com o método *wavelet* o algoritmo não identifica erroneamente as pessoas, ocorrendo apenas algumas rejeições.

Para o subconjunto 2 pode-se concluir que, sem a utilização de um método de correção de iluminação, o algoritmo identifica erroneamente, poucas vezes, as outras pessoas como sendo a pessoa 4, 5 e 6, e identifica erroneamente, apenas uma vez, a pessoa 3 como sendo a pessoa 1. Com o método variação do filtro homomórfico, o algoritmo identifica erroneamente, poucas vezes, as outras pessoas como sendo a pessoa 6, e identifica erroneamente, apenas uma vez, as outras pessoas como sendo a 1, 4, 5 e 10. Com o método LogAbout, o algoritmo identifica erroneamente, poucas vezes, outras pessoas como sendo a pessoa 5, o algoritmo também identifica erroneamente, apenas uma vez, a pessoa 3 com sendo a pessoa 1. Finalmente, com o método *wavelet*, o algoritmo não identifica erroneamente as pessoas, ocorrendo apenas um número alto de rejeições.

Para o subconjunto 3 pode-se concluir que, sem a utilização de um método de correção de iluminação, o algoritmo identifica erroneamente, poucas vezes, as outras pessoas como sendo a pessoa 1, 4, 5 e 10. Com o método variação do filtro homomórfico, o algoritmo identifica erroneamente, poucas vezes, as outras pessoas como sendo a pessoa 1, 5 e 10, e identifica erroneamente, apenas uma vez, outras pessoas como sendo a pessoa 4, 8 e 9. Com o método LogAbout, o algoritmo identifica erroneamente, poucas vezes, as outras pessoas como sendo a pessoa 1, 5 e 8, e identifica erroneamente, apenas uma vez, outras pessoas como sendo a pessoa 6 e 10. Finalmente, com o método *wavelet*, o algoritmo identifica erroneamente, apenas uma vez, a pessoa 2 como sendo a pessoa 1.

Conforme já mencionado, a matriz de confusão reforçou a conclusão de que o método *wavelet* sobressaiu-se em relação aos demais, seguido pelo método LogAbout.

6.2.2 Algoritmo utilizando *wavelets* e rede neural

O algoritmo utilizando *wavelets* e rede neural baseou-se no algoritmo de Rizk et al. (2004). Foi utilizada apenas a rede MLP desse algoritmo, uma vez que esta apresentou melhores resultados do que a Rede Neural de Base Radial (RBF).

Neste algoritmo foi aplicada a Transformada de *Wavelet*, usando as *wavelets* de Symlets 1, com três níveis de decomposição. Essas *wavelets* são similares às da família Daubechies, apresentando as mesmas propriedades, porém são quase simétricas, o que não ocorre com a família Daubechies. Obteve-se como resultado dessa decomposição, uma imagem 32x32, que

comparando-se com a não utilização de um determinado método. Avaliando-se os resultados, observa-se que houve pouca melhora para os três métodos de normalização de iluminação. O método com a variação do filtro homomórfico destacou-se por apresentar os piores resultados. Além disso, avaliando-se se foi vantajoso ou não o uso de um método de normalização para o reconhecimento facial, conclui-se que houve melhora na taxa de reconhecimento com o método LogAbout e *wavelet* para o subconjunto 3, e pouca melhora para o subconjunto 2, em ambos os casos. Não houve melhora para o subconjunto 1 com o método LogAbout e *wavelet*, e para os subconjuntos 1, 2 e 3 com o método variação do filtro homomórfico.

Figura 50: Gráfico do método que utiliza *wavelet* e MLP.

Tabela 5: Resultado do método que utiliza *wavelets* e RNA no subconjunto 3.

Métodos aplicados	% de melhora			
	Subconjunto 1	Subconjunto 2	Subconjunto 3	Média
LogAbout	-	8	12	6,6
Variação do filtro homomórfico	-	-	-	-
<i>Wavelet</i>	-	4	12	5,3

Os métodos de normalização da iluminação não apresentaram uma melhora considerável na taxa de reconhecimento com o algoritmo *wavelet* e MLP. Isso ocorre devido

ao fato da rede MLP não conseguir separar adequadamente características faciais com valores muito próximos (GASPAR, 2006). Os métodos de normalização da iluminação LogAbout e *wavelet* mostraram-se eficazes em melhorar imagens escuras (Figura 42 e Figura 46), e como o subconjunto 3 contém mais imagens desse tipo (Figura 38), ocorreram resultados satisfatórios na taxa de reconhecimento para este subconjunto.

Será considerada a seguir a avaliação através da matriz de confusão para o algoritmo de reconhecimento facial *wavelet* com MLP (anexo B, página 96), gerada para cada método de normalização da iluminação e gerada também para as imagens sem correção de iluminação, aplicados sobre os subconjuntos 1, 2 e 3. Na matriz de confusão foram consideradas as 10 pessoas utilizadas para teste, com cinco imagens para cada pessoa.

Para o subconjunto 1 pode-se concluir que, sem a utilização de um método de correção de iluminação, o algoritmo identifica erroneamente, uma única vez, a pessoa 7 como sendo a pessoa 5, e identifica erroneamente, duas vezes, a pessoa 8 como sendo a pessoa 3. Com o método variação do filtro homomórfico, o algoritmo identifica erroneamente, uma única vez, a pessoa 3 como sendo a pessoa 6. Com o método LogAbout, o algoritmo identifica erroneamente, uma única vez, a pessoa 1 como sendo a pessoa 3, e a pessoa 9 como sendo a pessoa 5. Finalmente, como o método *wavelet*, o algoritmo identifica erroneamente, uma única vez, a pessoa 1 como sendo a pessoa 3, e a pessoa 10 como sendo a pessoa 2.

Para o subconjunto 2 pode-se concluir que, sem a utilização de um método de correção de iluminação, o algoritmo identifica erroneamente, poucas vezes, as outras pessoas como sendo a pessoa 2, e identifica erroneamente, apenas uma vez, as outras pessoas como sendo a 6 e a 7. Com o método variação do filtro homomórfico, o algoritmo identifica erroneamente, apenas uma vez, as outras pessoas como sendo a 2, 5 e 9. Com o método LogAbout, o algoritmo identifica erroneamente, poucas vezes, as outras pessoas como sendo a pessoa 4. Finalmente, com o método *wavelet* o algoritmo identifica erroneamente, pouca vezes, as outras pessoas como sendo a pessoa 2, e identifica erroneamente, apenas uma vez, as outras pessoas como sendo a pessoa 6 e 7.

Para o subconjunto 3 o algoritmo não identifica erroneamente as pessoas, exceto para o método *wavelet*, que o algoritmo identifica erroneamente, uma única vez, a pessoa 2 como sendo a pessoa 4, e a pessoa 7 como sendo a pessoa 5.

Nos três subconjuntos o número de rejeições foi consideravelmente alta, o que confirma que a rede MLP não é capaz de separar adequadamente características faciais muito próximas.

6.2.3 Algoritmo utilizando auto-faces

No algoritmo de auto-faces apresentado em Serrano (2007), foram efetuadas pequenas modificações para a adequação da base de dados desse trabalho: não foi utilizada a normalização das imagens do algoritmo de Serrano, uma vez que o foco deste trabalho é comparar métodos de normalização da iluminação, e foi determinado um limiar para verificar se uma dada imagem de face estava ou não no banco. Neste algoritmo, foi determinado um único limiar para todas as pessoas da mesma base de dados durante a fase de treinamento, e na fase de teste foi verificado se cada face de entrada pertencia ou não ao banco.

O gráfico da Figura 51 ilustra a taxa de acerto do algoritmo de reconhecimento utilizando auto-faces, para os subconjuntos 1, 2 e 3. A Tabela 6 mostra a taxa de melhora do algoritmo de reconhecimento para cada método de normalização da iluminação, comparando-se com a não utilização da normalização. O símbolo “-” da tabela indica que não ocorreram melhoras nas taxa de reconhecimento, ou se obteve uma taxa inferior comparando-se com a não utilização de um determinado método. Pelos resultados apresentados, observa-se que os métodos LogAbout e *wavelets* obtiveram bons resultados, porém para este último não ocorreram melhoras para o subconjunto 2. Além disso, avaliando-se se foi vantajoso ou não o uso de um método de normalização para o reconhecimento facial, conclui-se que houve uma melhoria considerável para o subconjunto 1 e 2 com o método LogAbout, e para o subconjunto 1 e 3 com o método *wavelet*. Mas, houve pouca melhora com o método LogAbout para o subconjunto 3 e não ocorreram melhoras nos três subconjuntos para o método variação do filtro homomórfico.

Figura 51: Gráfico do método que utiliza auto-faces.

Tabela 6: Resultado do método que utiliza auto-faces.

Métodos aplicados	% de melhora à imagem original			
	Subconjunto 1	Subconjunto 2	Subconjunto 3	Média
LogAbout	18	12	4	11,3
Variação do filtro homomórfico	-	-	-	-
<i>Wavelet</i>	12	-	18	10

Na Figura 51, comparando-se os resultados da taxa de reconhecimento das imagens de face sem a utilização de um método de correção da iluminação com o método variação do filtro homomórfico, observa-se que são muito próximos. Já o método LogAbout conseguiu atingir bons resultados pois foi capaz de melhorar tanto as imagens claras quanto as escuras (Figura 42). O método *wavelets* não atingiu melhora na taxa do reconhecimento no subconjunto 2, pois este subconjunto é o que tem menos problemas de iluminação (como pode ser observado na Figura 38). Como o método auto-faces utiliza a imagem média de todas as imagens do banco, esta média tem um valor próximo das imagens que não utilizam um determinado método de correção da iluminação.

Será considerada a seguir a avaliação através da tabela com taxa de acerto de cada pessoa para o algoritmo de reconhecimento facial auto-faces (anexo C, página 103), gerada para cada método de normalização da iluminação e gerada também para as imagens sem correção de iluminação, aplicados sobre os subconjuntos 1, 2 e 3. Na tabela foram consideradas as 10 pessoas utilizadas para teste, com cinco imagens para cada pessoa.

Para o subconjunto 1 pode-se concluir que, sem a utilização de um método de correção de iluminação, o algoritmo obteve uma taxa de acerto de 20% apenas para a pessoa 9. Com o método *wavelet*, o algoritmo obteve uma taxa de acerto de 40% para a pessoa 9 e de 100% para a pessoa 10. Com o método LogAbout, o algoritmo obteve uma taxa de acerto de 100% para as pessoas 9 e 10. Finalmente, com o método variação do filtro homomórfico, o algoritmo obteve uma taxa de acerto de 20% para a pessoa 9.

Para o subconjunto 2 pode-se concluir que, sem a utilização de um método de correção de iluminação, o algoritmo obteve uma taxa de acerto de 0% para todas as pessoas. Com o método *wavelet*, o algoritmo obteve a mesma taxa de 0% para todas as pessoas. Com o método LogAbout, o algoritmo obteve uma taxa de 100% para a pessoa 2 e de 20% para a pessoa 5. Finalmente, com o método variação do filtro homomórfico, o algoritmo obteve uma taxa de acerto de 0% para todas as pessoas.

Para o subconjunto 3 pode-se concluir que, sem a utilização de um método de correção de iluminação, o algoritmo obteve uma taxa de acerto de 100% para a pessoa 5 e de 50% para a pessoa 8. Com o método *wavelet*, o algoritmo obteve uma taxa de acerto de 100% para a pessoa 1 e 3, uma taxa de acerto de 40% para a pessoa 4, uma taxa de acerto de 20% para a pessoa 8, e uma taxa de acerto de 60% para a pessoa 10. Com o método LogAbout, o algoritmo obteve uma taxa de acerto de 20% para as pessoas 4 e 8, uma taxa de acerto de 60% para a pessoa 5, e uma taxa de acerto de 80% para a pessoa 8. Finalmente, com o método variação do filtro homomórfico, o algoritmo obteve uma taxa de acerto de 80% para a pessoa 5, e de 40% para a pessoa 8.

A taxa de acerto médio das imagens sem normalização da iluminação e das imagens que utilizam o método variação do filtro homomórfico foram iguais, pois a taxa de acerto para cada pessoa nos subconjuntos 1 e 2 foram os mesmos, e a taxa de acerto de cada pessoa para o subconjunto 3 ficaram próximas.

7 Conclusões e sugestões para trabalhos futuros

7.1 Considerações iniciais

Nesse capítulo são apresentadas as conclusões, as contribuições desse trabalho e as propostas para trabalhos futuros.

7.2 Conclusões

Conforme mencionado na seção 1.1, entre os maiores desafios no reconhecimento facial está o problema de iluminação. Este trabalho efetuou uma comparação dos seguintes métodos de normalização da iluminação: LogAbout (LIU et al., 2002), variação do filtro homomórfico (DELAC et al., 2006), método baseado em *wavelets* (DU e WARD, 2005). O objetivo foi avaliar a porcentagem de melhoria no reconhecimento facial com o uso de cada método.

Os seguintes algoritmos de reconhecimento de faces foram utilizados para avaliar cada método: auto-faces (SERRANO, 2007), PCA com rede neural LVQ (FERNANDES e GONZAGA, 2007) e rede neural com *wavelets* (RIZK et al., 2004).

Analisando-se os resultados apresentados na seção 6.2, observa-se que o método de normalização com *wavelets* destacou-se para o algoritmo de reconhecimento PCA com rede neural LVQ, e obteve o segundo melhor resultado para os algoritmos auto-faces e *wavelets* com rede neural MLP, apesar de ter apresentado baixa porcentagem de melhoria para o último algoritmo.

O segundo método de normalização, o LogAbout, apresentou o segundo melhor resultado com o algoritmo PCA com rede neural LVQ, com melhorias significativas. Além disso, apresentou o melhor resultado para o algoritmo auto-faces e *wavelet* com rede neural MLP, apesar de ter apresentado uma baixa porcentagem de melhoria para o último algoritmo.

O terceiro método de normalização, variação do filtro homomórfico, foi o que mostrou um desempenho inferior em relação aos demais, apresentando uma baixa porcentagem de melhoria para os algoritmos de reconhecimento PCA com rede neural MLP, e não obtendo melhora para os demais algoritmos.

Considerando o tempo de processamento e a porcentagem de melhoria obtida para os três algoritmos de reconhecimento, o método de normalização *wavelet* e o LogAbout foram os que obtiveram os melhores resultados.

Com relação ao tempo de processamento, o LogAbout é o método que apresentou o menor tempo seguido de *wavelets*.

Os resultados também evidenciaram que, de uma maneira geral, com a utilização dos métodos de normalização da iluminação obtém-se uma melhor taxa do reconhecimento facial, exceto para o caso do método variação do filtro homomórfico, que apresentou apenas uma pequena melhora para o algoritmo de reconhecimento PCA com rede neural LVQ.

Um ponto interessante a ser destacado nesse trabalho é que as imagens que os classificadores conseguem reconhecer, não necessariamente são reconhecidas facilmente pelos seres humanos. Quando foi aplicado o algoritmo de reconhecimento PCA com LVQ sobre as imagens resultante do método de normalização que utiliza *wavelet* foram obtidos bons resultados, apesar destas imagens não poderem ser visualizadas de maneira satisfatória pelo ser humano (Figura 46).

7.3 Contribuições

Além dos resultados apresentados, este trabalho traz as seguintes contribuições:

- descrição de algoritmos de amenização da iluminação encontrados na literatura, que são fáceis de serem implementados;
- comprovação que o método de normalização *wavelet* e LogAbout podem ser utilizados para melhorar a taxa de reconhecimento, o que é uma informação importante para

quem deseja melhorar a iluminação de um imagem para, posteriormente, efetuar reconhecimento facial;

- evidenciou que nem sempre as imagens que são melhores para os seres humanos reconhecerem são as melhores para os algoritmos de reconhecimento facial.

- introdução teórica necessária para aqueles que desejam iniciar pesquisas na área de reconhecimento facial, uma vez que são apresentados desde métodos de pré-processamento até algoritmos de reconhecimento de faces, de uma maneira simples e fácil de serem compreendidos.

7.4 Sugestões para trabalhos futuros

Sugere-se como trabalho futuro uma avaliação detalhada com relação às direções da fonte de iluminação numa imagem, para identificar o método de normalização de iluminação que obtém melhores resultados com uma direção de iluminação específica. Nesse trabalho, as imagens foram separadas em relação ao grau de elevação da iluminação, ou seja, direção da luz na vertical, e seria interessante avaliar as imagens com relação ao grau azimute, direção da iluminação na horizontal.

O algoritmo *retinex* (seção 3.9) apresentou resultados satisfatórios quando aplicados em imagens com a finalidade de melhorar o contraste destas, porém este algoritmo ainda não foi utilizado com a finalidade de melhorar a taxa do reconhecimento facial, logo, sugere-se como trabalho futuro a utilização deste algoritmo para melhorar a iluminação em imagens de faces.

Uma vez que o método *wavelet* de Du e Ward (2005) obteve um dos melhores resultados nos algoritmos de reconhecimento facial, sugere-se como trabalho futuro a utilização de outro método que utiliza *wavelet* dos mesmos autores, o método adaptativo de realce de imagem baseado em região (DU e WARD, 2006). Uma das deficiências do método *wavelet* utilizado nesse trabalho é que ele é um método global, enquanto que o novo método de Du e Ward (2006) baseia-se na diferença de regiões locais com diferentes condições de iluminação, ou seja, o realce das imagens é realizado localmente em diferentes regiões da face, o que pode melhorar a taxa de reconhecimento obtida nesse trabalho.

Finalmente, sugere-se o uso de um algoritmo de validação para verificar a melhoria da qualidade das imagens, após utilizar as técnicas de normalização da iluminação.

Referências Bibliográficas

[TOOLBOX 2001] - MISITI, M.; MISITI, Y.; OPPENHEIM, G.; POGGI, J. M. “Wavelet Toolbox: User's Guide”, Version 2.1, MathWorks Inc., 2001.

ADINI, Y.; MOSES, Y.; ULLMAN, S. (1997). Face Recognition: The Problem of Compensating for Changes in Illumination Direction. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 721-732.

ARAIZA, E. A. R.; SANTIBAÑEZ, J. D. M.; RUIZ, G. H.; GUTIÉRREZ, C. A. G.; PEREA, M. T.; MORENO, G. J. R. (2007). Mejora de contraste y compensación en cambios de la illumination. *Computación y Sistemas*, vol. 10, pp. 357-371.

BAILEY, S. P. (2006). Neural Network Design on the SRC-6 Reconfigurable Computer. Dissertação (mestrado). Escola naval de pós-graduação, 2006.

BRAGA, A. de P.; CARVALHO, A. C. P. de L. F. de; LUDERMIR, T. B. (1998). *Fundamentos de Redes Neurais Artificiais*. São Paulo: Acadêmica.

BRUNELLI, R.; POGGIO, T. (1993). Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1042-1052.

CARVALHO, A. C. P. de L. F. de. Redes Neurais Artificiais. Disponível em: <<http://www.icmc.usp.br/~andre/research/neural/index.htm>>. Acesso em: 20/02/2007.

CASTRO, L. N. de (2003). Fundamentals of Neurocomputing. State University of Campinas. Technical Report.

CHEN, W.; ER, M. J.; WU, S. (2004). Illumination Compensation and Normalization Using Logarithm and Discrete Cosine Transform. *Proc. of IEEE International Conference on Control, Automation, Robotics and Vision*, vol. 1, pp. 380-385.

DAUBECHIES, I. (1990). The wavelet transform, time-frequency localization and signal analysis. *IEEE Transaction on Information Theory*, vol. 36, pp. 961-1005.

DELAC, K.; GRGIC, M.; KOS, T (2006). Sub-Image Homomorphic Filtering Techniques for Improving Facial Identification under Difficult Illumination Conditions. *In: Internacional Conference on System, Signals and Image Processing*. Budapest.

DU, S.; WARD, R. (2005). Wavelet-Based Illumination Normalization for Face Recognition. *In: IEEE International Conference on Image Processing*, vol. 2, pp. 954-957.

DU, S.; WARD, R. (2006). Adaptive Region-Based Image Enhancement Method for Face Recognition Under Varying Illumination Conditions. *In: IEEE International Conference on Acoustics, Speech and Signal Processing*, vol 2, pp.353-356.

FARIA, R. R. A. (1997). Aplicação de *Wavelets* na Análise de Gestos Musicais em Timbres de Instrumentos Acústicos Tradicionais. Dissertação (Mestrado em Engenharia de Sistemas Eletrônicos) - Escola de Politécnica da Universidade de São Paulo, 2008.

FERNANDE, R. A. S.; GONZAGA, A. (2007). Reconhecimento de Faces Utilizando Redes Neurais Artificiais com Fusão de Características Faciais por Método de Votação. *III Workshop de Visão Computacional*, São José do Rio Preto - SP, pp. 296-300.

GASPAR, T. L. (2006). Reconhecimento de Faces Humanas Usando Redes Neurais MLP. 134 p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos. 2006.

GEORGHIADES, A. S.; BELHMEURS, P. N.; KRIEGMAN, D. J. (2000). From Few To Many: Generative Models For Recognition Under Variable Pose and Illumination. *IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pp. 277-284, 2000.

GONZALEZ, R. C.; WOODS, R. E. (2002). *Digital Image Processing*. Editora Prentice Hall.

HAYKIN, S. (2001). *Redes Neurais Artificiais: princípios e prática*. Editora Bookman.

HERNANDEZ, E.; WEISS, G. (1996). *A First Course on Wavelets*. CRC Press.

HEUSCH, G.; CARDINAUX, F.; MARCEL, S. (2005). Lighting Normalization Algorithms for Face Verification. IDIAP Technical Report.

HUANG, S.-J.; HSIEH, C.-T. (2002). Coiflet wavelet transform applied to inspect power system disturbance-generated signals. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, pp. 204-210.

JOBSON, D. J.; RAHMAN, Z.; WOODDELL, G. A. (1997). Properties and Performance of a Center/Surround Retinex. *IEEE Transactions on Image Processing*, vol. 6, pp. 451-462.

Kitani E. C e Thomaz C. E. (2006). Um Tutorial sobre Analise de Componentes Principais para o Reconhecimento Automático de Faces. Technical Report, Departamento de Engenharia Elétrica, FEI, São Bernardo do Campo, São Paulo, Brazil.

LAND, E.; MACCANN, J (1971). Lightness and Retinex Theory. *Journal of the Optical Society of America*, nº 61.

LEVINE, M. D.; GANDHI, M. R.; BHATTACHARYYA J. (2004). Image Normalization for Illumination Compensation in Facial Images. Department of Electrical & Computer Engineering & Center for Intelligent Machines, McGill University, Montreal, Canada, Unpublished Report.

LIU, D.; LAM, K.M.; SHEN, L.S. (2004). Optimal sampling of Gabor features for face recognition. *Pattern Recognition Letters*, vol. 25, pp. 267-276.

LIU, H.; MIAO, J.; LI, J. et al (2002). A Novel Method to Compensate Variety of Illumination in Face Detection. In: *6th Joint Conference on Information Sciences (ICCVPRIP)*, pp 692-695.

LUCCA, E. V. D. (2006). Avaliação e Comparação de Algoritmos de Segmentação de Imagens de Radar de Abertura Sintética. Dissertação (Doutorado) Instituto Nacional de Pesquisas Espaciais - INPE, 2006.

MALLAT, S. (1999). *A Wavelet Tour of Signal Processing (2 ed.)*. Academic Press.

MARQUES FILHO, O.; VIEIRA NETO, H. (1999). *Processamento Digital de Imagem*. Rio de Janeiro. Editora Brasport.

MEYER, Y. (1993). *Wavelets: Algorithms and Applications*. Society for Industrial and Applied Mathematic.

MITRA, S.; SAVVIDES, M. (2005). Gaussian mixture models based on the frequency spectra for human identification and illumination classification. *Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, pp. 245-250.

MOSES, Y.; ULLMAN, S. (1992). Limitation of Non-Model-Based Recognition Schemes. Springer, vol. 588, pp. 820-828.

PENHARBEL, E. A.; WUTZKE, E. L. G.; SILVA, M. dos S.; BIANCHI, R, A. C. (2005). E-Faces - Um classificador capaz de analisar imagens e classificá-las como faces ou não faces utilizando o método Eigenfaces. *I Workshop de Visão Computacional*, Piracicaba, SP, pp. 13-19.

PENTLAND, A.; MOGHADDAM, B.; STARNER, T. (1994). View-based and modular eigenspaces for face recognition. *In Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pp. 84-91.

PIZZER, S. M.; AMBURN, E. P.; AUSTIN, J. D.; CROMARTIE, R.; GESELOWITZ, A.; GREER, T.; ROMENY, B. T. H.; ZIMMERMAN, J. B. (1987). Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, vol. 30, pp. 355-368.

QIAO, F. (2005). Introduction to Wavelet - A tutorial. *Workshop 118 on Wavelet Application in Transportation Engineering*.

REIGOTA, N. dos S. (2007). Comparação da Transformada Wavelet discreta e da Transformada do Cosseno para Compressão de Imagens de Impressão Digital. Dissertação (Mestrado em Engenharia Elétrica) - Escola de Engenharia de São Carlos, 2007. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18152/tde-27042007-101810>>. Acesso em: 01/03/2007.

RAHMAN, Z.; JOBSON, D. J.; WOODDELL, G (1996). A. Multi-scale retinex for color image enhancement. *In: International Conference on Image Processin*, vol. 3, pp. 1003-1006.

RAHMAN, Z., WOODDELL, G. A.; Jobson; D. J. (1997). A Comparison of the Multiscale Retinex With Other Image Enhancement Techniques. *Proceedings of the IS&T 50th Anniversary Conference*.

RIZK, M. R. M.; SAID, O.; EL-SAYED, R.; SOBHY, K. (2004). Wavelets and neural networks based face recognition system. *Proceedings of the 46th IEEE International Midwest Symposium on Circuits and Systems*, vol. 2, pp. 965-968.

SANTOS, A. R. dos (2005). Identificação de faces humanas através de PCA-LDA e redes neurais SOM. 149 p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos. 2005.

SERRANO, S. Eigenface Tutorial. Disponível em: <<http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial.htm>>. Acesso em 06/08/2007.

SHAN, S.; GAO, W.; CAO, B.; ZHAO, D. (2003). Illumination Normalization for Robust Face Recognition Against Varying Lighting Conditions. *Proc. of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, vol. 17, pp. 157-164.

SOARES, A. da S. (2008). Predição de Séries Temporais Econômicas por meio de Redes Neurais Artificiais e Transformada Wavelet: Combinando Modelo Técnico e

Fundamentalista. Dissertação (Mestrado em Engenharia Elétrica) - Escola de Engenharia de São Carlos, 2008.

STRICKERT, M. (2004). Analysis of LVQ in the context of spontaneous egg signal classification. Dissertação (Mestrado em ciência da computação) - Universidade do Bielefeld, 2004.

TAN, X.; CHEN, S.; ZHOU, Z.; ZHANG, F. (2006). Face recognition from a single image per person: A survey. *The Journal of Pattern Recognition Society*, vol. 39, pp. 1725-1745.

THULASIRAMAN, P. (2005). Semantic Classification of Rural and Urban Images Using Learning Vector Quantization. Dissertação (Mestrado em engenharia industrial) - Universidade do estado da Louisiana, 2005.

VIEIRA, A. de F. (2006). Identificação de relâmpagos e sprites na atmosfera pelo imageador ALIS a bordo do satélite científico EQUARS. Dissertação (Mestrado em Ciência da Computação) - Instituto de Ciências Matemáticas e de Computação (ICMC), 2006.

ZHANG, Z.; MA, S.; WU, D. (2005). The Application of Neural Network and Wavelet in Human Face Illumination Compensation. *Springer*, pp. 828-835.

ZHAO, W.; CHELLAPPA, R.; PHILLIPS, P. J.; ROSENFELD, A. (2003). Face Recognition: A Literature Survey. *ACM Computing Surveys*, vol. 35, No. 4, pp. 399–458.

ZHU, J.; LIU, B.; SCHWARTZ, S. C. (2003). General Illumination Correction and its Application to Face Normalization. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 133-6.

Anexo A – Matriz de confusão com os resultados do algoritmo de reconhecimento facial PCA com LVQ

Nesse anexo é apresentada a taxa de acerto, taxa de erro, taxa de rejeição e a matriz de confusão para os algoritmos de normalização da iluminação em cada subconjunto de faces utilizando o algoritmo de reconhecimento facial PCA com LVQ. A Tabela 7 mostra a taxa de reconhecimento, que ficou igual a 16%, além de mostrar a taxa de erro e de rejeição. A Tabela 8, por sua vez, mostra a matriz de confusão, onde as linhas indicam as classes de entrada (cada pessoa) e as colunas indicam as classificações feitas, para essa classe de entrada. Considere a linha 4, representando a classe 4. A coluna 4 com o valor 2 indica 2 classificações corretas. A coluna 6 com o valor 3 indica que 3 padrões da classe 2 foram classificados como pertencente à classe 6. A coluna rejeição, com o valor 0 indica que todos os padrões da classe 2 foram classificados.

Resultados do subconjunto 1

Tabela 7: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 1 original.

Acertos	8	Erros	33	Rejeições	9
Taxa de acerto	16%	Taxa de erro	66%	Taxa de rejeição	18%

Tabela 8: Tabela de confusão com os resultados do reconhecimento com o algoritmo PCA com LVQ aplicados sobre o subconjunto 1 original.

Pessoa/ classificação	1	2	3	4	5	6	7	8	9	10	Rejeição
1	0			2		2					1
2		0		1		3					1
3	1		0	1		3					0
4				2		3					0
5					2	3					0
6				3		2					0
7						1	0				4
8						2		0			3
9						4			1		0
10				3		1				1	0

Tabela 9: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método *wavelets*.

Acertos	31	Erros	0	Rejeições	19
Taxa de acerto	62%	Taxa de erro	0%	Taxa de rejeição	38%

Tabela 10: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método *wavelets*.

Pessoa/ classificação	1	2	3	4	5	6	7	8	9	10	Rejeição
1	3										2
2		2									3
3			5								0
4				2							3
5					3						2
6						4					1
7							2				3
8								3			2
9									3		2
10										4	1

Tabela 11: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.

Acertos	24	Erros	15	Rejeições	11
Taxa de acerto	48%	Taxa de erro	30%	Taxa de rejeição	22%

Tabela 12: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.

Pessoa/ classificação	1	2	3	4	5	6	7	8	9	10	Rejeição
1	3									1	1
2		0							2		3
3			3			2					0
4				3						2	0
5					2					2	1
6						4				1	0
7							2			1	2
8								2		1	2
9									3		2
10				1	1				1	2	0

Tabela 13: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.

Acertos	9	Erros	40	Rejeições	1
Taxa de acerto	18%	Taxa de erro	80%	Taxa de rejeição	2%

Tabela 14: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.

Pessoa/ classificação	1	2	3	4	5	6	7	8	9	10	Rejeição
1	0			1						4	0
2		1								3	1
3	1		0	1						3	0
4				2						3	0
5					2					3	0
6				2		0				3	0
7							0			5	0
8								0		5	0
9									1	4	0
10				2						3	0

Resultados do subconjunto 2

Tabela 15: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 2 original.

Acertos	6	Erros	8	Rejeições	36
Taxa de acerto	12%	Taxa de erro	16%	Taxa de rejeição	72%

Tabela 16: Tabela de confusão com os resultados do reconhecimento com o algoritmo PCA com LVQ aplicados sobre o subconjunto 2 original.

Pessoa/ classificação	1	2	3	4	5	6	7	8	9	10	Rejeição
1	1										4
2		1									4
3	1		0			2					2
4				3							2
5					1						4
6				1	1	0					3
7							0				5
8								0			5
9									0		5
10				2	1					0	2

Tabela 17: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método *wavelet*.

Acertos	21	Erros	0	Rejeições	29
Taxa de acerto	42%	Taxa de erro	0%	Taxa de rejeição	58%

Tabela 18: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método *wavelet*.

Pessoa/ classificação	1	2	3	4	5	6	7	8	9	10	Rejeição
1	2										3
2		1									4
3			3								2
4				2							3
5					3						2
6						3					2
7							2				3
8								1			4
9									2		3
10										2	3

Tabela 19: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.

Acertos	17	Erros	23	Rejeições	10
Taxa de acerto	34%	Taxa de erro	46%	Taxa de rejeição	20%

Tabela 20: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.

Pessoa/ classificação	1	2	3	4	5	6	7	8	9	10	Rejeição
1	2					1					2
2		2				2					1
3	1		0		1	2					1
4				2		1					2
5					3	2					0
6					2	3					0
7							2				3
8						4		1			0
9						2			2		1
10					3	2				0	0

Tabela 29: Taxa de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.

Acertos	16	Erros	22	Rejeições	12
Taxa de acerto	32%	Taxa de erro	44%	Taxa de rejeição	24%

Tabela 30: Tabela de confusão com os resultados de reconhecimento do algoritmo PCA com LVQ aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.

Pessoa/ classificação	1	2	3	4	5	6	7	8	9	10	Rejeição
1	2				1						2
2	2	0			1						2
3			1		1			1		1	1
4				2	1					1	1
5					3						2
6	1				1	0			1	2	0
7					1		2			2	0
8					1			2			2
9					1				2		2
10				1	2					2	0

Tabela 33: Taxa de reconhecimento do algoritmo *wavelets* com MLP aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método *wavelet*.

Acertos	31	Erros	2	Rejeições	17
Taxa de acerto	62%	Taxa de erro	4%	Taxa de rejeição	34%

Tabela 34: Tabela de confusão com os resultados de reconhecimento do algoritmo *wavelets* com MLP aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método *wavelet*.

[illegible]

Tabela 35: Taxa de reconhecimento do algoritmo *wavelets* com MLP aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.

Acertos	36	Erros	3	Rejeições	11
Taxa de acerto	72%	Taxa de erro	6%	Taxa de rejeição	22%

Tabela 36: Tabela de confusão com os resultados de reconhecimento do algoritmo *wavelets* com MLP aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.

[illegible]

Tabela 41: Taxa de reconhecimento do algoritmo *wavelets* com MLP aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método *wavelet*.

Acertos	22	Erros	6	Rejeições	22
Taxa de acerto	44%	Taxa de erro	12%	Taxa de rejeição	44%

Tabela 42: Tabela de confusão com os resultados de reconhecimento do algoritmo *wavelets* com MLP aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método *wavelet*.

[illegible]

Tabela 43: Taxa de reconhecimento do algoritmo *wavelets* com MLP aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.

Acertos	24	Erros	2	Rejeições	24
Taxa de acerto	48%	Taxa de erro	4%	Taxa de rejeição	48%

Tabela 44: Tabela de confusão com os resultados de reconhecimento do algoritmo *wavelets* com MLP aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.

[illegible]

Tabela 49: Taxa de reconhecimento do algoritmo *wavelets* com MLP aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método *wavelet*.

Acertos	38	Erros	2	Rejeições	10
Taxa de acerto	76%	Taxa de erro	4%	Taxa de rejeição	20%

Tabela 50: Tabela de confusão com os resultados de reconhecimento do algoritmo *wavelets* com MLP aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método *wavelet*.

[illegible]

Tabela 51: Taxa de reconhecimento do algoritmo *wavelets* com MLP aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.

Acertos	38	Erros	0	Rejeições	12
Taxa de acerto	76%	Taxa de erro	0%	Taxa de rejeição	24%

Tabela 52: Tabela de confusão com os resultados de reconhecimento do algoritmo *wavelets* com MLP aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.

[illegible]

Anexo C – Matriz de confusão com os resultados do algoritmo de reconhecimento facial auto-faces

Neste anexo é apresentada a taxa de acerto para cada pessoa de cada subconjunto de faces utilizando o algoritmo de reconhecimento auto-face.

Resultados do subconjunto 1

Tabela 55: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-face aplicados sobre o subconjunto 1 original.

Pessoa	Taxa de acerto (em porcentagem)
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	20
10	0
Total	2

Tabela 56: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método *wavelet*.

Pessoa	Taxa de acerto (em porcentagem)
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	40
10	100
Total	14

Tabela 57: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.

Pessoa	Taxa de acerto (em porcentagem)
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	100
10	100
Total	20

Tabela 58: Tabela com taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 1 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.

Pessoa	Taxa de acerto (em porcentagem)
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	20
10	0
Total	2

Resultados do subconjunto 2

Tabela 59: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-face aplicados sobre o subconjunto 2 original.

Pessoa	Taxa de acerto (em porcentagem)
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
Total	0

Tabela 60: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método *wavelet*.

Pessoa	Taxa de acerto (em porcentagem)
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
Total	0

Tabela 61: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.

Pessoa	Taxa de acerto (em porcentagem)
1	0
2	100
3	0
4	0
5	20
6	0
7	0
8	0
9	0
10	0
Total	12

Tabela 62: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 2 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.

Pessoa	Taxa de acerto (em porcentagem)
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
Total	0

Resultados do subconjunto 3

Tabela 63: Tabela com a taxa de acerto para cada pessoa do reconhecimento com o algoritmo auto-face aplicados sobre o subconjunto 3 original.

Pessoa	Taxa de acerto (em porcentagem)
1	0
2	0
3	0
4	0
5	100
6	0
7	0
8	50
9	0
10	0
Total	14

Tabela 64: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método *wavelet*.

Pessoa	Taxa de acerto (em porcentagem)
1	100
2	0
3	100
4	40
5	0
6	0
7	0
8	20
9	0
10	60
Total	32

Tabela 65: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método LogAbout.

Pessoa	Taxa de acerto (em porcentagem)
1	0
2	0
3	0
4	20
5	60
6	0
7	0
8	80
9	20
10	0
Total	18

Tabela 66: Tabela com a taxa de acerto para cada pessoa do algoritmo auto-faces aplicados sobre o subconjunto 3 após a aplicação do algoritmo de normalização que utiliza o método variação do filtro homomórfico.

Pessoa	Taxa de acerto (em porcentagem)
1	0
2	0
3	0
4	0
5	80
6	0
7	0
8	40
9	0
10	0
Total	12