



Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Programa de Pós-Graduação em Engenharia Elétrica e
Computação



Sistema Não-Intrusivo para Estimação da Direção do Olhar Utilizando Redes Neurais Artificiais

Helton Maia Peixoto

Orientadora: Prof^ª. Dra. Ana Maria Guimarães Guerreiro

Co-orientador: Prof. Dr. Adrião Duarte Dória Neto

Dissertação de Mestrado apresentada
ao Programa de Pós-Graduação em En-
genharia Elétrica e Computação da UFRN
(área de concentração: Engenharia de
Computação) como parte dos requisitos
para obtenção do título de Mestre em
Ciências.

Número de ordem PPgEEC: M000
Natal, RN, Novembro de 2010

Seção de Informação e Referência

Catálogo da publicação na fonte. UFRN / Biblioteca Central Zila Mamede

Peixoto, Helton Maia.

Sistema Não-Intrusivo para Estimação da Direção do Olhar Utilizando Redes Neurais Artificiais/ Helton Maia Peixoto - Natal, RN, 2010
64 f.: il.

Orientadora: Ana Maria Guimarães Guerreiro

Co-orientador: Adrião Duarte Dória Neto

Dissertação (mestrado) - Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica e de Computação.

1. Processamento Digital de imagens - Dissertação. 2. Redes Neurais Artificiais - Dissertação. I. Guerreiro, Ana Maria Guimarães. II. Dória Neto, Adrião Duarte. III. Universidade Federal do Rio Grande do Norte. IV. Título.

RN/UF/BCZM

CDU 004.383.8

Sistema Não-Intrusivo para Estimação da Direção do Olhar Utilizando Redes Neurais Artificiais

Helton Maia Peixoto

Dissertação de Mestrado aprovada em 26 de novembro de 2010 pela banca examinadora composta pelos seguintes membros:

Prof^a. Dra. Ana Maria Guimarães Guerreiro (orientadora) . DCA/UFRN

Prof. Dr. Adrião Duarte Dória Neto (co-orientador) DCA/UFRN

Prof^a. Dra. Heliana Bezerra Soares (examinador interno) ... DCA/UFRN

Prof. Dr. Celso Alberto Saibel Santos (examinador externo) DCC/UFBA

*Aos meus pais e familiares, com
carinho.*

Agradecimentos

A minha orientadora e ao meu co-orientador, professores Ana Maria Guimarães Guerreiro e Adrião Duarte Dória Neto, sou grato pela orientação.

Aos meus pais, familiares e amigos, por todo carinho, amor, apoio, confiança e credibilidade.

A minha namorada Rossana Moreno, pelo amor, compreensão, paciência e incentivo incondicional nas minhas escolhas.

Aos demais colegas de pós-graduação, pelas críticas e sugestões.

Resumo

Os sentidos fundamentais do corpo humano são: visão, audição, tato, gustação (ou paladar) e olfato. Estes constituem as funções que propiciam o nosso relacionamento com o ambiente. A visão funciona como um receptor sensorial responsável pela captação de informações do mundo exterior que serão enviadas ao cérebro. O olhar reflete a sua atenção, intenção e interesse. Sendo assim, a estimação da direção do olhar, a partir de modelos computacionais, possibilita uma alternativa promissora para melhorar a capacidade de interação homem-máquina, inclusive dos portadores de deficiências motoras. O objetivo deste trabalho consiste em apresentar um sistema não-intrusivo que utiliza basicamente um computador pessoal e uma *webcam* de baixo custo que, quando aliados ao uso das técnicas de processamento digital de imagens, transformadas *Wavelets* e reconhecimento de padrões, com as redes neurais artificiais, resultam em um sistema completo que realiza desde a aquisição de imagens (passando pela detecção da face e rastreamento dos olhos) até a estimação da direção do olhar. Os resultados apresentados mostram a viabilidade do sistema proposto assim como diversas vantagens em seu uso.

Palavras-chave: Estimação da Direção do Olhar, Processamento Digital de Imagens, Redes Neurais Artificiais.

Abstract

The fundamental senses of the human body are: vision, hearing, touch, taste and smell. These senses are the functions that provide our relationship with the environment. The vision serves as a sensory receptor responsible for obtaining information from the outside world that will be sent to the brain. The gaze reflects its attention, intention and interest. Therefore, the estimation of gaze direction, using computer tools, provides a promising alternative to improve the capacity of human-computer interaction, mainly with respect to those people who suffer from motor deficiencies. Thus, the objective of this work is to present a non-intrusive system that basically uses a personal computer and a low cost webcam, combined with the use of digital image processing techniques, Wavelets transforms and pattern recognition, such as artificial neural network models, resulting in a complete system that performs since the image acquisition (including face detection and eye tracking) to the estimation of gaze direction. The obtained results show the feasibility of the proposed system, as well as several feature advantages.

Keywords: Estimation of Gaze Direction, Digital Image Processing, Artificial Neural Networks.

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	v
Lista de Siglas e Abreviaturas	vii
1 Introdução	1
1.1 Motivação	3
1.2 Objetivos	3
1.3 Estrutura do trabalho	4
2 Processamento de Imagens	7
2.1 Estrutura do olho humano	7
2.2 Sistemas Intrusivos <i>versus</i> Não-Intrusivos	8
2.3 Espectro de cores	10
2.3.1 O modelo RGB de cores	10
2.3.2 Conversão para níveis de cinza	11
2.4 Segmentação de imagens	11
2.4.1 Limiarização de imagens	12
2.4.2 Limiarização de Otsu	13
2.5 Detecção da Face	13
2.6 Centro de massa	14
2.7 Transformadas <i>Wavelets</i>	15
2.7.1 Vantagens de utilização	15
2.7.2 <i>Wavelet</i> discreta	17
2.7.3 Análise multiresolução	18
2.7.4 Compressão	20
2.8 Conclusão	22
3 Redes Neurais Artificiais	24
3.1 Processos de aprendizagem	25
3.2 Rede perceptrons de múltiplas camadas (MLP)	26
3.3 Algoritmos de treinamento	28
3.3.1 Algoritmo de retropropagação (<i>backpropagation</i>)	28
3.3.2 Algoritmo <i>resilient backpropagation</i> (RPROP)	29

3.3.3	Algoritmos de gradiente conjugado	30
3.3.4	Algoritmos baseados no método Quasi-Newton	31
3.3.5	Algoritmo Levenberg-Marquardt (LM)	32
3.4	Conclusão	33
4	Sistema Não Intrusivo para Estimação da Direção do Olhar	35
4.1	Processamento de baixo nível	36
4.1.1	Capturando e pré-processando imagens	36
4.2	Processamento de nível intermediário	37
4.2.1	Segmentação das imagens	37
4.2.2	Construindo o banco de dados	40
4.3	Processamento de alto nível	41
4.3.1	Redes neurais: treino e classificação	41
4.4	Conclusão	42
5	Análise de Resultados	45
5.1	Segmentando imagens	45
5.1.1	Detectando a face	45
5.1.2	Detectando o olho	50
5.2	Estimando com redes neurais artificiais	50
5.2.1	Conjunto de treinamento: Extraíndo descritores	51
5.2.2	Estimando a direção do olhar	53
5.3	Conclusão	56
6	Conclusão	58
	Referências bibliográficas	61

Lista de Figuras

1.1	Etapas fundamentais para o processamento de imagens.	4
2.1	Fisiologia do olho humano: (http://biofisicahumana.blogspot.com). . .	8
2.2	Equipamentos especiais usados para a interação homem-máquina através da visão: (http://www.nttdocomo.com/features/mobility18.html). . .	9
2.3	Câmera de captura composta por diodos emissores de luz infravermelha: (http://www.gizmowatch.com/entry/eyebow2-an-eye-for-billboards-to-count-their-viewers/).	9
2.4	Representação gráfica do cubo RGB.	10
2.5	Figura clássica da Lena.	11
2.6	Histograma bimodal.	12
2.7	Imagem e seu centro de massa.	15
2.8	Análise espectral de sinais, comparando as transformadas de Fourier e <i>Wavelet</i>	16
2.9	Função de escala e <i>Wavelet</i> primária de Haar.	19
2.10	Decomposição da <i>wavelet</i>	19
2.11	Representação multiresolução de uma imagem.	20
2.12	Análise multiresolução de uma imagem.	20
2.13	Representação da compressão de uma imagem.	21
3.1	Neurônios biológicos.	24
3.2	Neurônio artificial.	25
3.3	Arquitetura genérica de uma rede neural artificial do tipo MLP com duas camadas ocultas.	27
3.4	Aprendizado supervisionado.	27
4.1	Diagrama de blocos do sistema desenvolvido.	35
4.2	Cenário utilizado para a captura das imagens.	37
4.3	Detecção da face.	38
4.4	Delimitando a área do olho.	38
4.5	Identificação de objetos dentro da área do olho.	39
4.6	Detecção do olho.	39
4.7	Identificação do centro de massa do olho.	39
4.8	Processo de construção do banco de dados.	40
4.9	Processamento de alto nível.	41
5.1	Detecção da face utilizando a métrica distância.	46

5.2	Detecção da face utilizando a métrica orientação espacial RIP.	47
5.3	Detecção da face utilizando a métrica orientação espacial ROP.	47
5.4	Detecção da face utilizando a métrica luminosidade.	48
5.5	Detecção da face utilizando a métrica diversidade.	49
5.6	Exemplos de detecção da face no ambiente proposto.	49
5.7	Exemplos de detecção do olho no ambiente proposto.	50
5.8	Amplitude de cada <i>pixel</i> em escala de cinza da imagem de detecção do olho.	51
5.9	Descritores das imagens: histogramas e coeficientes <i>Wavelets</i>	52
5.10	Erro médio quadrático em função do tempo de processamento para treinamento da rede em cada algoritmo.	53

Lista de Tabelas

5.1	Resultados de aplicação da métrica distância.	46
5.2	Resultados de aplicação da métrica orientação espacial.	47
5.3	Resultados de aplicação da métrica luminosidade.	48
5.4	Resultados de aplicação da métrica diversidade.	49
5.5	Exemplos de detecção do olho.	50
5.6	Vetor de descritores.	52
5.7	Matriz de confusão para a rede MLP-1.	54
5.8	Matriz de confusão para a rede MLP-1.	55
5.9	Matriz de confusão para a rede MLP-2.	55

Lista de Siglas e Abreviaturas

BFGS - Broyden-Fletcher-Goldfarb-Shanno

CGB - *Powell-Beale Conjugate Gradient Backpropagation*

CGP - *Conjugate Gradient with Powell-Beale Restarts*

CIE - *Commission Internationale d'Eclairage*

CMYK - *Cyan, Magenta, Yellow and Black*

db - Daubechies

DCA - Departamento de Engenharia da Computação e Automação

DWT - *Discrete Wavelet Transform*

FCC - *Federal Communications Commission*

JPEG - *Joint Photographic Experts Group*

LM - Levenberg-Marquardt

MLP - Rede Perceptrons de Múltiplas Camadas

MSE - *Mean Square Error*

PC - *Personal Computer*

RAM - *Random Access Memory*

RGB - *Red, Green and Blue*

RIP - *Rotation in Plane*

RNA - Rede Neural Artificial

ROP - *Rotation on Plane*

RPROP - *Resilient Backpropagation*

SCG - *Scaled Conjugate Gradient*

SMQT - *The Successive Mean Quantization Transform*

STFT - *Short Time Fourier Transform*

UFRN - Universidade Federal do Rio Grande do Norte

YCbCr - *Luminance (Y), Chrominance Blue (Cb), Chrominance Red (Cr)*

Capítulo 1

Introdução

Com o desenvolvimento tecnológico dos últimos anos, os computadores cada vez mais fazem parte do cotidiano de milhões de pessoas. Desenvolver recursos para a interação homem-máquina é uma maneira prática de possibilitar a inclusão social dessas pessoas, melhorando consideravelmente a qualidade de suas vidas. São as necessidades de um mundo moderno, aliadas à possibilidade de preencher lacunas do desenvolvimento, que montam uma base para a aplicação dos conhecimentos científicos em benefício da humanidade. Hoje a Engenharia, de forma geral, possibilita tal desenvolvimento.

O rastreamento dos olhos é a técnica usada para detectar a posição para onde a pessoa está olhando. Seu conceito baseia-se em focar o olho do usuário e de alguma forma estimar a direção do seu olhar. Além dos vários conceitos relacionados à matemática, também é preciso entender um pouco da fisiologia humana, mais especificamente os assuntos relacionados à visão, para abordar esse problema de uma forma satisfatória. Este trabalho concatena diversas linhas de pesquisa que envolvem os sistemas inteligentes atuais, tais como processamento de imagens, transformadas *Wavelets* e redes neurais artificiais, tendo em vista reconhecer padrões e classificar a direção do olhar.

Entre os cinco sentidos fundamentais do corpo humano (visão, audição, tato, gustação ou paladar e olfato), a visão é o que envia o maior volume de informações para o cérebro. Assim, a estimação da direção do olhar é uma alternativa promissora para a comunicação homem-máquina, além de ser uma forma de comunicação natural e rápida.

Desde o século XIX já existia o interesse em estudar os movimentos dos olhos. Mais precisamente em 1879, o oftalmologista francês nascido em Paris, Louis Émile Javal, mesmo desprovido das tecnologias atuais para certos experimentos, observou que a leitura não envolve uma varredura suave dos olhos ao longo do texto, como anteriormente se pensava, mas uma série de curtas paradas e sacadas rápidas. Edmund Huey (1968) construiu um rastreador ocular precoce, utilizando uma espécie de lente de contato com um buraco para a pupila.

O primeiro sistema não-intrusivo foi construído por Guy Thomas Buswell (1935) em Chicago, utilizando o reflexo de feixes de luz nos olhos dos usuários. O psicólogo russo Alfred Lukjanovic Yarbus (1967) apresentou, em livro, uma série de estudos relacionados ao monitoramento ocular, estabelecendo concordância com Buswell,

quanto ao fato de que os padrões de inspeção visual do ambiente dependem tanto de propriedades do estímulo quanto dos objetivos prévios do observador. Na década de 1970, os estudos sobre rastreamento ocular tiveram uma rápida expansão; uma visão bem estruturada sobre o assunto foi concebida por Rayner (1978). A década de 1980 foi marcada por várias pesquisas sobre a relação de atividades cerebrais e o olho humano, como [CARPENTER 1980, POSNER 1980, WARD 1980]. A partir da década de 1990, foram surgindo cada vez mais motivações como, por exemplo, o *hardware* computacional eficiente e as novas técnicas do processamento digital de imagens. Isto garantiu uma maior precisão nos estudos do rastreamento dos olhos e despertou uma maior atenção dos pesquisadores [HOFFMAN 1998, SCHNEIDER 1996, MULLIGAN 1997, DAUNICHT 1992].

Nos últimos 10 anos, diversas possibilidades no Brasil e no resto do mundo surgiram para a estimação da direção do olhar. Estas podem ser realizadas tanto no espaço 3D, pela determinação de um vetor correspondente à linha de visão do usuário, quanto em 2D, determinando o ponto observado sobre uma superfície de interesse em particular (um painel ou a tela do monitor, por exemplo) [FLICKNER 2000]. Vários sistemas que permitem a interação homem-máquina através da visão humana já foram desenvolvidos. Entretanto, para o rastreamento dos olhos, a maioria requer o uso de equipamentos especiais como eletrodos instalados próximos aos olhos, lentes de contato e capacetes [MORETTO 2004, NETO 2003]. Tais sistemas são conhecidos como intrusivos, uma vez que os aparelhos entram em contato direto com a pele ou olho do usuário, o que pode provocar certo desconforto e restringir o seu tempo de utilização [MORETTO 2004]. Tipicamente, procuram-se nas imagens pontos característicos como a pupila, a íris, a esclera ou mesmo reflexos gerados por fontes de luz, os quais são usados como referência na determinação da direção do olhar [FLICKNER 2000]. Normalmente, nesses sistemas, são usadas câmeras com diodos emissores de luz infravermelha, que facilitam a detecção das pupilas humanas, e câmeras filmadoras com foco automático. Apesar de fornecerem bons resultados, tais equipamentos possuem um custo elevado, dificultando o acesso à utilização [MORETTO 2004]. Nos sistemas não-intrusivos, técnicas baseadas em visão computacional contornam certas limitações estimando a direção do olhar a partir do processamento de imagens capturadas da face ou do olho das pessoas [OGORMAN 1995, GOMES 2001, TAVARES 2000]. Um cronograma detalhado da evolução destes sistemas recentemente foi publicado por Dan Witzner Hansen e Qiang Ji, o qual pode ser visto em [JI 2010].

Este trabalho faz parte do contexto dos sistemas não-intrusivos e pode ser visto como um modelo que contempla técnicas robustas de otimização (processamento de imagens e reconhecimento de padrões) com o objetivo de estimar a direção do olhar. Algumas condições de desenvolvimento geram dificuldades diversas e são comumente evitadas pela maioria dos sistemas intrusivos e não-intrusivos. Neste trabalho, tais condições são levadas em consideração, como premissas de desenvolvimento. Várias destas premissas são apresentadas no decorrer do trabalho, mas uma merece atenção prévia: a utilização de um *hardware* (*webcam*) para a captura das imagens, simplificado e sem qualquer controle de luminosidade ou auxílio de *leds* de

qualquer natureza, que facilitem a detecção de objetos. Isto gera um desafio ainda maior na busca de bons resultados.

1.1 Motivação

A principal motivação para a elaboração desta proposta de trabalho é a possibilidade de contribuição para uma melhor interação homem-máquina, mais especificamente na estimação da direção do olhar, um campo de estudo que se desenvolve constantemente e está diretamente ligado às novas abordagens e possibilidades computacionais que surgem a cada dia. Uma enorme evolução dos sistemas inteligentes e do poder de processamento dos computadores fornece alternativas antes não possíveis, mas que hoje podem ser efetivamente desenvolvidas.

A estimação da direção do olhar permite um universo de aplicações, onde a criatividade é o limite. Hoje em dia, os dispositivos móveis como aparelhos celulares, *notebooks*, *handhelds* e outros, todos equipados com câmeras, são uma constante nos ambientes domésticos. Propõe-se então, que tais dispositivos sejam controlados através do movimentos do globo ocular, para que pessoas portadoras de deficiências, como é o caso das portadoras da síndrome de *locked-in* (doença neurológica rara, caracterizada pela paralisia completa dos músculos voluntários em todas as partes do corpo, exceto os que controlam o movimento dos olhos), possam se comunicar com o mundo exterior, selecionar e ativar um ícone, um item de menu na tela do computador para registrar o interesse do usuário por determinado conteúdo de uma página da *Internet*, entre outros benefícios. Motoristas podem se comunicar com periféricos dentro de seus automóveis e podem até ser monitorados por câmeras que detectem níveis de atenção e sonolência como medida de segurança. Isto torna ainda mais possível a interatividade homem-máquina.

Por fim, a motivação do trabalho pode ser sintetizada pelo desafio de proporcionar alguma melhoria nas abordagens metodológicas para a estimação da direção do olhar; algo que esteja compatível com as condições de trabalho existentes e disponíveis à realização da pesquisa.

1.2 Objetivos

Este trabalho tem como principal objetivo o desenvolvimento de um sistema não-intrusivo, que utiliza um *hardware* de baixo custo aquisitivo (uma simples *webcam*) acoplado a um computador. O sistema deve ser de fácil manipulação e possibilitar a interação homem-máquina a partir da detecção da direção do olhar proveniente do movimento dos olhos.

O desenvolvimento de um sistema pode ser visto como uma coletânea de estruturas e recursos que estão interligados por uma lógica que proporcione o alcance de um ou mais objetivos.

Assim, utilizando mais especificamente, técnicas de processamento digital de imagens, extração de características com transformadas *Wavelets* e reconhecimento

de padrões com as redes neurais artificiais, é possível estimar a direção do olhar satisfatoriamente, compreendendo um sistema completo, que vai desde a aquisição das imagens até o reconhecimento de padrões para estimação da direção do olhar.

Deste modo, para estimar a direção do olhar, será apresentado um conjunto de técnicas empregadas por sistemas inteligentes que, aliadas a outras técnicas desenvolvidas neste trabalho, compõem mais uma relevante alternativa de organização e execução de recursos computacionais físicos e lógicos para a resolução do problema exposto.

1.3 Estrutura do trabalho

Neste e em muitos trabalhos de processamento de imagens, um sistema pode ser dividido em três etapas essenciais. São elas: processamento de baixo nível, que compreende a aquisição e o pré-processamento das imagens; processamento de nível intermediário, com objetivo de analisar as imagens; processamento de alto nível, para o reconhecimento e a interpretação de padrões, que são tarefas ligadas à cognição inteligente.

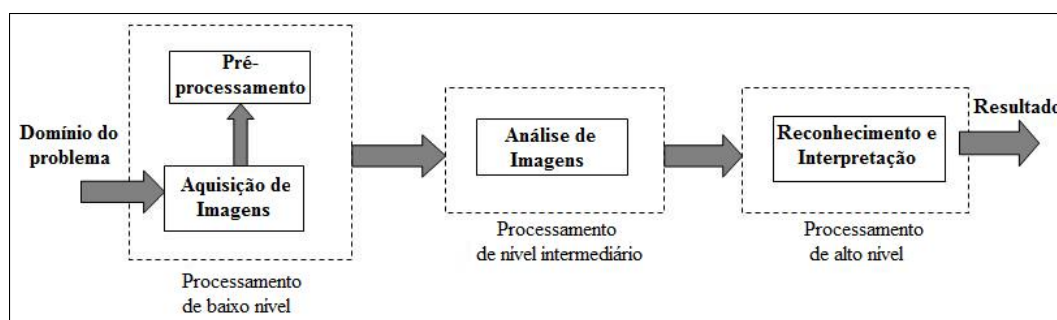


Figura 1.1: Etapas fundamentais para o processamento de imagens.

Baseado na Figura 1.1 e obedecendo a uma ordem lógica de organização, este trabalho tem seu conteúdo dividido em seis capítulos, que podem ser sumariamente descritos como:

Capítulo 2: neste capítulo, há a exposição geral da fundamentação teórica do processamento digital de imagens, ressaltando os principais conteúdos utilizados no trabalho. Este embasamento sobre o processamento de imagens possibilita entender o funcionamento de diversos algoritmos utilizados e a motivação para que sejam implementados. Tal conteúdo está presente tanto na fase de processamento de baixo nível quanto no processamento de nível intermediário. Este capítulo objetiva a extração de características relevantes das informações, as quais servirão de base para obtenção dos resultados desejados.

Capítulo 3: neste capítulo, é abordado o conteúdo referente ao reconhecimento e à interpretação das informações trabalhadas nas etapas anteriores. Esta etapa é

chamada de processamento de alto nível. O capítulo 3 relaciona os conceitos de redes neurais artificiais para a classificação de padrões, que constitui a etapa final para a estimação da direção do olhar.

Capítulo 4: neste capítulo, o enfoque é dado ao sistema proposto. Tal sistema é resultado da organização de conceitos e idéias apresentadas nos capítulos anteriores, entre eles: processamento de imagens, transformadas *Wavelets* e redes neurais artificiais. Com a construção do sistema, torna-se possível iniciar a etapa de implementações, objetivando um enfoque mais prático do trabalho.

Capítulo 5: este capítulo apresenta a evolução dos resultados mais relevantes do trabalho, incluindo algumas publicações realizadas na área. Nele, também são validadas as metodologias e técnicas desenvolvidas. Os resultados são analisados detalhadamente, com a finalidade de possibilitar um entendimento completo do que foi proposto neste trabalho.

Capítulo 6: este capítulo apresenta as conclusões gerais do trabalho e ressalta as principais contribuições. Além da precisão dos resultados, é verificada a importância de um sistema particionado, que permite melhorias e a execução de vários testes independentes, proporcionando um ganho enorme de produtividade. Finalmente, são feitas algumas sugestões para a continuidade da pesquisa.

Capítulo 2

Processamento de Imagens

Este capítulo tem como objetivo abordar conceitos importantes relacionados ao processamento digital de imagens, sendo esses adequados ao objetivo final de estimação da direção do olhar.

Técnicas e inovações surgem a cada momento no campo da computação gráfica, em áreas como: síntese de imagens, processamento de imagens e visão computacional. Premissas importantes para o desenvolvimento destas áreas estão associadas à otimização dos recursos físicos de *hardware* e ao custo computacional necessário. Trabalhar com imagens nesse contexto exige a junção de duas linhas do conhecimento: a teoria matemática de composição e manipulação das imagens e o desenvolvimento de programas(algoritmos) que possam assim implementar tais modelos.

As próximas seções fornecem o conteúdo essencial para o entendimento do sistema proposto neste trabalho, exibindo o funcionamento dos principais algoritmos utilizados para o processamento das imagens e suas aplicações.

2.1 Estrutura do olho humano

É importante um breve entendimento dos mecanismos de funcionamento do sistema visual humano, principalmente quando se trabalha com a captura e processamento de imagens. Sistemas computacionais tentam imitar o olho humano, tanto na aquisição de imagens, quanto em seu processamento e armazenamento. Este trabalho utiliza alguns destes recursos artificiais, como uma câmera para obtenção das imagens e o posterior reconhecimento de padrões visuais.

O olho humano pode ser considerado uma câmera superautomática capaz de posicionar-se na direção do objeto de interesse, focalizando e regulando sua sensibilidade à luminância do ambiente a partir de receptores especializados que se encontram na retina [LENT 2002]. O olho é semelhante a uma esfera com um diâmetro médio de cerca de 20 *mm*. Na Figura 2.1, observam-se algumas partes importantes: a esclera é a parte branca do olho que serve para protegê-lo; a córnea é uma estrutura transparente e resistente que permite a passagem da luz para dentro do olho e ajuda a focalizá-la na retina; a íris é a parte que dá cor aos olhos, sua função é controlar o tamanho da pupila, por meio da contração e expansão de seus músculos; a pupila é uma abertura na íris que aumenta ou diminui, controlando a

quantidade de luz que penetra no olho de acordo com a luminosidade do ambiente; o cristalino é uma lente biconvexa que auxilia na focalização da imagem sobre a retina, funcionando como um fino ajuste para o foco e a leitura; por fim, a retina é responsável pela transmissão das imagens recebidas pelo cérebro, através do nervo óptico [GONZALEZ 2000].

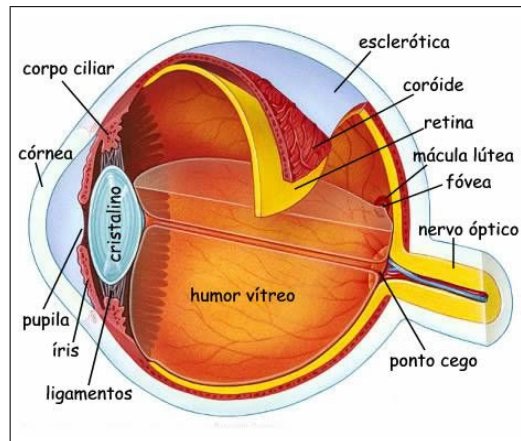


Figura 2.1: Fisiologia do olho humano: (<http://biofisicahumana.blogspot.com>).

Quando o olhar é direcionado para algum objeto, a imagem atravessa a córnea e chega à íris, que regula a quantidade de luz recebida por meio da pupila. Neste momento, quanto maior estiver a pupila, mais luz entra no olho. Passando pela pupila, a imagem chega ao cristalino, e é focada sobre a retina. A lente do olho produz uma imagem invertida, e o cérebro a converte para a posição correta.

Para o ser humano, a visão de cores começa na retina onde acontece uma relação de sensibilidade dos cones à certos comprimentos de onda da luz. Assim, uma complexa combinação pode ser estabelecida e ligada a um padrão de resposta dos neurônios responsáveis pela visão [LENT 2002]. Na retina, mais de cem milhões de células fotorreceptoras transformam sinais luminosos em impulsos eletroquímicos, que são enviados e decodificados pelo cérebro.

2.2 Sistemas Intrusivos *versus* Não-Intrusivos

A aquisição de imagens é normalmente realizada utilizando um sensor de imageamento, como por exemplo, uma câmera que tem como finalidade representar a imagem real do usuário em um formato digital que será convenientemente processado pelo computador.

A imagem analógica (real) é representada por uma função intensidade de luz bidimensional $f(x,y)$, onde x e y denotam as coordenadas espaciais e o valor da função f representa o brilho da imagem naquele ponto. Uma imagem digital é discreta tanto nas coordenadas espaciais quanto no brilho. Matematicamente, tem-se uma matriz cujos índices de linhas e colunas identificam um ponto na imagem e o valor desse ponto exibe seu respectivo brilho [GONZALEZ 2000].

Dependendo do dispositivo (sensor) de captura de imagens utilizado, os sistemas para encontrar a direção do olhar podem ser classificados como intrusivos ou não-intrusivos. Os sistemas intrusivos empregam dispositivos especiais como óculos e eletrodos para a aquisição das imagens dos olhos. Os sistemas não-intrusivos empregam câmeras nesse processo de aquisição. Segundo Moretto (2004), as técnicas intrusivas possuem uma taxa de acerto da localização dos olhos maior que os sistemas não-intrusivos, uma vez que a localização do olho é facilitada. Entretanto, técnicas não-intrusivas são de difícil acesso e possuem custo aquisitivo elevado, inviabilizando seu uso.



(a) Óculos (Human Interface Technology Laboratory da Universidade de Washington, DC.)

(b) Eletrooculograma.

Figura 2.2: Equipamentos especiais usados para a interação homem-máquina através da visão: (<http://www.nttdocomo.com/features/mobility18.html>).



Figura 2.3: Câmera de captura composta por diodos emissores de luz infravermelha: (<http://www.gizmowatch.com/entry/eyebox2-an-eye-for-billboards-to-count-their-viewers/>).

Alguns sistemas não-intrusivos (ver Figuras 2.2 e 2.3) utilizados para detectar a direção do olhar fazem uso de um processo prévio de ajuste de valores para cada usuário, chamado de calibração [JI 2010]. Esses valores sofrem modificações a cada interação dos diferentes usuários, como por exemplo, a distância do usuário com relação à câmera e a luminosidade do ambiente [MORETTO 2004].

2.3 Espectro de cores

A compreensão do espectro de cores é de fundamental importância quando se trabalha com captura e processamento de imagens. Com este conhecimento, torna-se possível tomar decisões principalmente a respeito dos padrões de cores, facilitando todo o restante do projeto a ser desenvolvido.

Dispositivos de entrada utilizados para captura de imagens como *scanners*, câmeras digitais, monitores de computador e outros fazem uso de um determinado modelo para a manipulação de cores. Existem vários padrões para representar as cores dos *pixels*, são exemplos: RGB (vermelho, verde, azul), HSB (matiz, saturação, brilho), CMYK (ciano, magenta, amarelo e preto) e CIE $L^*a^*b^*$ (L sendo para luminosidade e os valores de a^* e b^* para dois componentes cromáticos)[KEESE 2009].

2.3.1 O modelo RGB de cores

Este trabalho escolheu utilizar o modelo RGB de cores. Este modelo é o mais encontrado em utilização por monitores coloridos e câmeras de vídeo. Através dele, cada cor aparece nos seus componentes espectrais primários: vermelho, verde e azul. Baseada em coordenadas cartesianas, a Figura 2.4 mostra bem esse subespaço de cores.

As imagens que utilizam o modelo RGB possuem, em geral, 3 canais de cores que podem reproduzir até 16,7 milhões de cores. Elas são conhecidas também como imagens de 24 *bits*, por serem compostas por canais de 8 *bits* de informações de cores por *pixel*. Cada *pixel* possui 256 possibilidades diferentes de vermelho, verde e azul.[KEESE 2009].

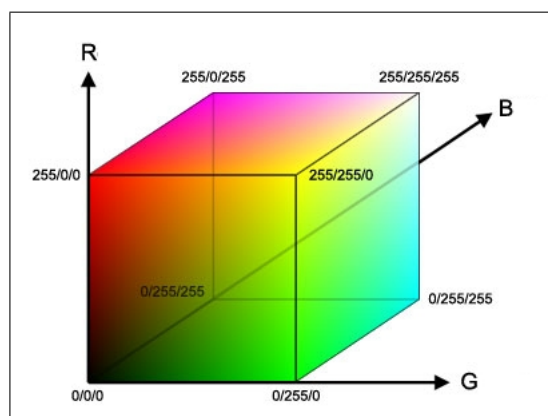


Figura 2.4: Representação gráfica do cubo RGB.

Os valores das componentes R, G e B são expressos nos eixos correspondentes e a soma vetorial destas componentes define um ponto de cor no espaço. Por isso, este modelo é chamado de modelo aditivo.

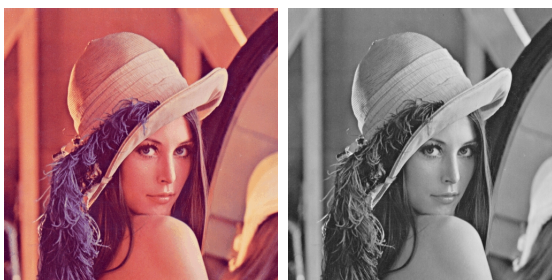
2.3.2 Conversão para níveis de cinza

Normalmente, a imagem capturada por uma câmera é colorida e salva no formato de cores RGB apresentado na seção 2.3.1. A imagem no formato RGB dificulta o processamento, uma vez que na detecção das bordas é provável que ocorra realce irrelevante em certas partes da imagem, como a pele, por exemplo. Isso ocorre porque as intensidades nos planos R, G e B serão alteradas diferentemente, resultando na mudança das intensidades relativas entre elas [FONSECA 2000, GONZALEZ 2000]. Sendo assim, é necessária a conversão da imagem para níveis de cinza.

Na conversão para níveis de cinza ocorre uma translação das cores originais para níveis intermediários entre o preto, que está na origem do plano, e o branco, que é o valor mais distante dessa origem. A conversão pode ser feita de acordo com (2.1), cuja execução deve ser realizada para cada *pixel* da imagem.

$$P'_{(x,y)} = P_{verm(x,y)} * 0,299 + P_{verd(x,y)} * 0,587 + P_{azul(x,y)} * 0,114 \quad (2.1)$$

Em (2.1), P' é o ponto da nova imagem em níveis de cinza e P_{verm} , P_{verd} e P_{azul} são os pontos da imagem original RGB [GONZALEZ 2000]. A Figura 2.5 é um exemplo dessa conversão *pixel a pixel* da clássica imagem da Lena, tanto no formato RGB quanto em níveis de cinza.



(a) Formato RGB. (b) Níveis de cinza.

Figura 2.5: Figura clássica da Lena.

2.4 Segmentação de imagens

Segmentar uma imagem significa subdividi-la em várias partes ou em seus objetos constituintes. Essa segmentação deve ser realizada até que os objetos de interesse tenham sido separados, lembrando que cada aplicação vai necessitar de uma técnica de segmentação em específico.

2.4.1 Limiarização de imagens

Neste trabalho, a limiarização (*thresholding*) é utilizada para converter imagens que estão em níveis de cinza para imagens binárias, sendo esta técnica também conhecida como binarização. Sabe-se que uma imagem no espaço vetorial RGB utiliza 8 *bits* para cada canal de informação, totalizando 24 *bits* por *pixel*. Para imagens em níveis de cinza, é comum o uso de um canal com 8 *bits* para cada *pixel*, totalizando 256 níveis possíveis de cinza. Já para as imagens binárias, necessita-se de apenas de um *bit* por *pixel*, totalizando duas cores: preto ou branco.

Usualmente, a limiarização consiste em separar as regiões de interesse de uma imagem através da escolha de um limiar (*threshold*). Tal processo pode ser visualizado na Figura 2.6, através da distribuição dos níveis de cinza de uma certa imagem representada pelo seu histograma. Os maiores picos devem conter a região de tonalidade mais clara e os menores picos contêm a tonalidade mais escura. Desta maneira, a determinação do limiar de corte dos níveis de cinza é a escolha de um valor ótimo entre estes picos.

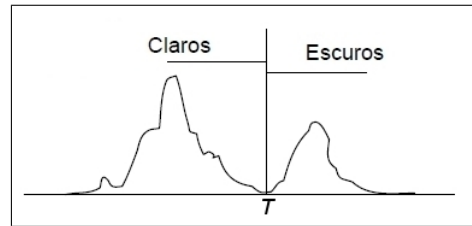


Figura 2.6: Histograma bimodal.

Ainda analisando o histograma da Figura 2.6, tem-se uma imagem $f(x,y)$ composta por objetos iluminados sobre um fundo escuro, de maneira que os *pixels* do objeto e os do fundo tenham seus níveis de cinza agrupados em dois grupos dominantes. A forma mais rápida de extrair tais objetos do fundo é através de um limiar T que separe os dois grupos. Então, cada *pixel* tal que $f(x,y) > T$ é considerado um *pixel* do objeto; caso contrário, um *pixel* do fundo. A limiarização pode ser vista matematicamente como uma função T dada por (2.2) [GONZALEZ 2000]:

$$T = T[x,y,f(x,y)] \quad (2.2)$$

em que $f(x,y)$ é o nível de cinza do *pixel*. Assim, uma imagem limiarizada pode ser definida como:

$$g(x,y) = \begin{cases} 1, & \text{se } f(x,y) > T \\ 0, & \text{se } f(x,y) \leq T \end{cases} \quad (2.3)$$

na qual os *pixels* com valor 1 correspondem aos objetos e os *pixels* com valor 0 correspondem ao fundo.

2.4.2 Limiarização de Otsu

O método de limiarização bimodal de Otsu é baseado na análise do histograma da imagem. Considerando l níveis de cinza em duas classes, C_0 e C_1 , que representam o objeto e o fundo, obtêm-se $C_0 = \{0, 1, \dots, t\}$ e $C_1 = \{t+1, t+2, \dots, l\}$. Seja σ_W^2 a variância dentro da classe, σ_B^2 a variância entre as classes e σ_T^2 a variância total e, de acordo com [OTSU 1979], a busca do limiar pode ser realizada a partir da minimização de umas das funções dadas em (2.4):

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2}, \quad \eta = \frac{\sigma_B^2}{\sigma_T^2} \quad \text{ou} \quad \kappa = \frac{\sigma_T^2}{\sigma_W^2} \quad (2.4)$$

2.5 Detecção da Face

O cérebro humano pode reconhecer padrões de forma trivial quando comparado com o mesmo processo sendo realizado artificialmente. Intuitivamente, as pessoas aprendem a reconhecer padrões relacionados aos sentidos do corpo humano (audição, olfato, paladar, tato e visão) desde que nascem. Isto é possível devido à capacidade do sistema nervoso, responsável por organizar e gerenciar todas as ações e processos de aprendizagem, à medida que o ser humano evolui. Desta forma, imitar este sistema artificialmente é de fato tarefa bastante complexa.

Hoje, o reconhecimento artificial da face é utilizado principalmente em aplicações como as que utilizam interfaces homem-máquina. Um bom exemplo seria o de um sistema para controle automático de acesso, onde é feita a aquisição da imagem do usuário e a mesma é comparada com outra imagem previamente armazenada em um banco de dados de faces. Deste modo, torna-se possível identificar quem está solicitando o acesso, para que este seja permitindo ou não.

O processo de detecção da face consiste em separar, a partir de uma imagem inicial, as informações da face propriamente ditas das demais informações, ou seja, criar uma classe contendo as faces e uma outra contendo informações de fundo. Esta não é uma tarefa tão simples porque, embora existam semelhanças entre as faces, estas semelhanças podem variar consideravelmente em função de outros parâmetros, como idade, cor da pele, orientação, expressão facial, oclusões, condições de iluminação, entre outros. É importante lembrar que o nível de controle dessas variáveis é extremamente relevante e está relacionado com o grau de complexidade do problema.

Diversas possibilidades podem ser utilizadas para resolver o problema de detecção da face. As principais técnicas de detecção utilizadas estão dispostas em quatro categorias [LOPES 2005]:

Baseadas em Conhecimento: são técnicas que utilizam conhecimentos a priori sobre aspectos faciais. Por exemplo, a maioria das faces possuem: dois olhos, um nariz e uma boca e ainda de certa forma, obedecem a padrões de geometria facial.

Um bom exemplo é o trabalho de Ming-Hsuan Yang [AHUJA 2002] que utiliza esse tipo de recurso para detecção da face.

Baseadas em Características Invariantes: em contraste com o método anterior, essa técnica identifica características faciais invariantes à orientação, como a cor da pele e a textura da face. Desta forma, é possível separar a face do restante dos objetos presentes em uma imagem.

Baseadas em *Templates*: técnicas bastante utilizadas na detecção de objetos. São, de certa forma, semelhantes às técnicas baseadas em conhecimento, onde a utilização de componentes geométricos básicos como círculos, triângulos ou quadrados serve para a criação de *templates*, que serão comparados com o objeto de interesse dentro da imagem. Heurísticas diversas, inclusive algoritmos genéticos, podem adicionar recursos à procura destes objetos. Pesquisadores como Alattar e Lee fazem uso desta técnica [RAJALA 1999][HONG 2001].

Baseadas na Aparência: nesta técnica, não se tem nenhum conhecimento a priori sobre as características da face que se deseja detectar. Este método utiliza principalmente algoritmos de aprendizagem onde as informações da face a ser detectada são aprendidas a partir de um treinamento. Uma base de dados com faces diversas é normalmente utilizada para tal treinamento. No trabalho de Brimblecombe (2009) tem-se a utilização das redes neurais artificiais; já em [ARA 1998], são utilizados os modelos ocultos de Markov.

Entre as diversas possibilidades para a detecção da face, a utilização da SMQT (*The Successive Mean Quantization Transform*) nesta tarefa chamou bastante atenção, principalmente pela compatibilidade apresentada em experimentos preliminares que podem ser vistos no capítulo 5, o que contribui para ser aplicada neste trabalho. Controlar variáveis ligadas à iluminação e variações do sensor (câmera de captura) para a detecção de objetos é de fato uma grande preocupação, inclusive quando se tem um ambiente diversificado. Assim sendo, esta abordagem é compatível com o estado da arte sobre a detecção de faces [CLAESSON 2005][CLAESSON 2007].

2.6 Centro de massa

Na área de processamento de imagens muitas vezes é comum ter que calcular centróides, baricentros ou o centro de massa das imagens. O centro de massa C de uma imagem (conjunto de *pixels*) é definido como a média das suas posições, r_i , ponderada por suas massas, m_i :

$$C = \frac{\sum m_i r_i}{\sum m_i} \quad (2.5)$$

A partir de (2.5), pode-se compor um simples exemplo da obtenção do centro de massa de uma imagem. A Figura 2.7 mostra que dentro desta imagem foi selecionada uma área delimitada pela linha de cor verde e dentro desta delimitação foi calculado seu centro de massa, que está indicado pelo ponto em azul.

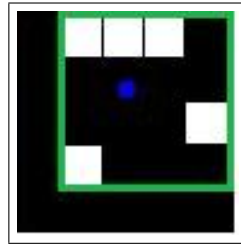


Figura 2.7: Imagem e seu centro de massa.

2.7 Transformadas *Wavelets*

As transformações bidimensionais desempenham um papel essencial no processamento digital de imagens. Uma série de trabalhos teóricos e aplicações foram desenvolvidas desde as últimas décadas até a atualidade. Dentre as transformadas mais utilizadas no processamento de imagens estão as transformadas de: Fourier, Walsh, Hadamard, cosseno discreto, Slant, Hotelling e *Wavelets*.

As *Wavelets* têm gerado enorme interesse dos cientistas, resultando no desenvolvimento de aplicações nas mais diversas áreas como: visão computacional [WANG 2009], sismologia [PAROLAI 2008], radar [MASNADI-SHIRAZI 2009], astronomia [OTTENSAMER 2008], compressão de imagens [BHATIA 2009], filtragem de sinais [VIMAL 2009], otimização de sistemas [PINTO 2009] e diversas outras. De forma geral, a grande vantagem do uso desta transformada é a possibilidade de aplicá-la a sinais não estacionários, o que permite o estudo do comportamento local das funções, não só no domínio da frequência, como também no domínio tempo-escala.

2.7.1 Vantagens de utilização

Métodos tradicionais de análise de sinal, baseados na transformada de Fourier, podem determinar todas as frequências presentes no sinal, porém sua relação com o domínio temporal inexiste. Para superar este problema, surgiu a transformada de Gabor (ou STFT - *Short Time Fourier Transform*): a idéia principal desta transformada é introduzir um novo parâmetro de frequência local como se a “transformada local” observasse o sinal através de uma curta janela dentro da qual o sinal permanece aproximadamente estacionário [OLIVEIRA 2007]. Os problemas na resolução do tempo e da frequência são resultados de um fenômeno físico conhecido como princípio da incerteza de Heisenberg (em um sinal, é impossível conhecer a frequência e o tempo exatos de ocorrência desta). Este fenômeno é indiferente em relação à transformada utilizada [OLIVEIRA 2007].

A transformada *Wavelet* foi desenvolvida como uma alternativa à transformada de Gabor para solucionar o problema da resolução. As *Wavelets* são funções matemáticas que separam sinais em diferentes componentes e extraem cada componente com uma resolução adequada à sua escala.

Obedecendo a composição das transformadas, tem-se a transformada de Fourier Contínua, que pode ser expressa na forma:

$$F(w) = \int_{-\infty}^{+\infty} f(t)e^{-j2\pi ft} dt \quad (2.6)$$

Conhecendo-se o espectro $F(w)$ de um sinal, é possível obtê-lo no domínio temporal, utilizando a sua transformada inversa, de acordo com (2.7):

$$f(t) = \int_{-\infty}^{+\infty} \frac{1}{2\pi} F(w)e^{j2\pi ft} dw \quad (2.7)$$

já a transformada *Wavelet* contínua pode ser expressa na forma:

$$CWT(\tau, a) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{+\infty} f(t)\psi\left(\frac{t-\tau}{a}\right) dt \quad (2.8)$$

e a sua inversa pode ser representada conforme (2.9):

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} CWT(a, b) \left\{ \frac{1}{\sqrt{a}} \psi\left(\frac{t-\tau}{a}\right) \right\} \frac{dad b}{a^2} \quad (2.9)$$

em que $\psi(t)$ é a *Wavelet* mãe, τ e a são os parâmetros translação e escala, respectivamente.

A Figura 2.8 fornece a análise espectral através das transformadas de Fourier e *Wavelet* de dois sinais [PROTÁZIO 2002].

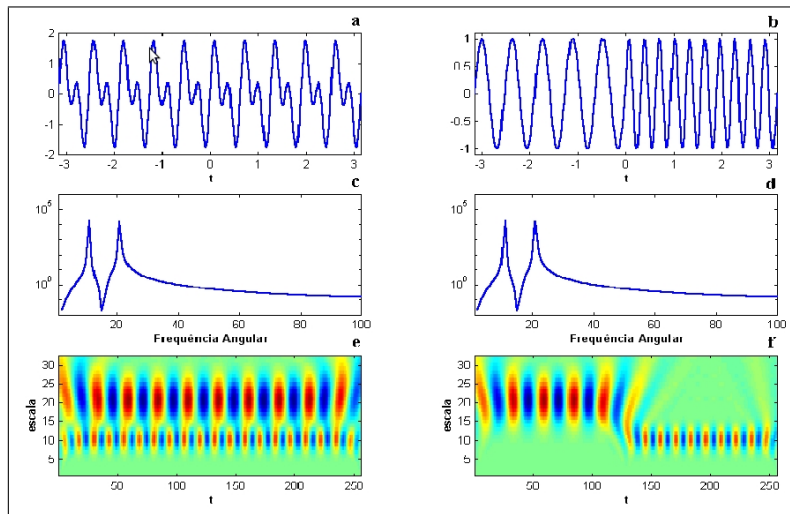


Figura 2.8: Análise espectral de sinais, comparando as transformadas de Fourier e *Wavelet*.

O primeiro sinal, mostrado na Figura 2.8(a), consiste da superposição de duas frequências ($\text{sen}10t$ e $\text{sen}20t$), e o segundo consiste das mesmas frequências aplicadas a cada uma das metades da duração do sinal, de acordo com a Figura 2.8(b). As Figuras 2.8 (c) e (d) mostram os espectros dos dois sinais obtidos através da transformada de Fourier aplicada aos sinais mostrados nas Figuras 2.8 (a) e (b), respectivamente. As Figuras (e) e (f) mostram a magnitude da transformada *Wavelet* dos mesmos sinais.

A partir da Figura 2.8, percebe-se a grande vantagem das *Wavelets* na análise de sinais não estacionários, traduzida na sua capacidade de fazer distinção entre estes sinais, ou seja, de localizar espacialmente no sinal cada uma das frequências envolvidas no mesmo.

2.7.2 *Wavelet* discreta

A transformada *Wavelet* contínua é calculada fazendo translações e escalonamentos contínuos de uma função sobre um sinal. Na prática, esta transformação não é viável, pois requer a realização de infinitas translações e escalonamentos, demandando muito tempo, esforço computacional e redundância. As *Wavelets* discretas foram introduzidas para superar este obstáculo e por isso serão utilizadas na realização do presente trabalho. As mesmas não são transladas nem escalonadas continuamente, mas sim em intervalos discretos, o que é conseguido a partir de uma modificação na *Wavelet* contínua:

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{|s|}} \Psi\left(\frac{t-\tau}{s}\right) \quad (2.10)$$

$$\Psi_{j,k}(t) = \frac{1}{\sqrt{|s_0^j|}} \Psi\left(\frac{t-k\tau_0 s_0^j}{s_0^j}\right) \quad (2.11)$$

em que j e k são inteiros; $s_0 > 1$ é um parâmetro de dilatação fixo; τ_0 é o fator de translação, que depende do fator de dilatação.

Geralmente, escolhe-se $s_0 = 2$ para que se tenha uma amostragem da frequência chamada de amostragem diádica e $\tau_0 = 1$ é escolhido para a amostragem temporal, também diádica. Isto resulta na expressão (2.12) [OLIVEIRA 2007]:

$$\Psi_{j,k}(t) = \sqrt{2^j} \Psi(2^j t - k) \quad (2.12)$$

Quando são usadas *Wavelets* discretas para analisar um sinal, o resultado é uma série de coeficientes *Wavelet*, também chamada de série de decomposição de *Wavelet* [OLIVEIRA 2007]. Como uma *wavelet* pode ser vista como um filtro passa-faixa,

a série de *Wavelets* escalonadas pode ser vista como um banco de filtros passa-faixa com fator Q (fator de fidelidade do banco de filtros). Na prática, tem-se uma *Wavelet* discretizada, com limites superiores e inferiores para as translações e os escalonamentos. As *Wavelets* discretizadas, aliadas à idéia de passar o sinal por um banco de filtros, resultam na conhecida codificação em sub-bandas (*Subband Coding*) [OLIVEIRA 2007].

2.7.3 Análise multiresolução

As transformadas matemáticas são utilizadas em um conjunto de dados para se obter informações adicionais não disponíveis no modelo de dados primitivos. No caso deste trabalho, é necessário o uso de uma transformada que detecte a variação dos tons de cores da vizinhança de um *pixel* e a localização espacial deste, e transponha isto de forma eficiente em um espaço multiresolução [CASTLEMAN 1996]. A análise multiresolução utilizando transformada *Wavelet* ganhou bastante popularidade com o lançamento do padrão JPEG-2000 [WEEKS 2007].

A análise multiresolução é uma estratégia de processamento de sinais onde é utilizado um conjunto de filtros especializados em extrair as informações de sinal, como as frequências nele presentes e a localização das mesmas em função do tempo de duração do sinal, em diferentes resoluções [CASTLEMAN 1996].

Uma breve descrição da análise multiresolução permite apresentar as duas funções responsáveis pela geração de todo o sistema de *Wavelets*: a função escala e a *Wavelet* primária (ou *Wavelet* mãe). O termo mãe vem do fato de que funções com diferentes tamanhos são usadas no processo da transformada e todas são originadas de uma *Wavelet* principal: *Wavelet* mãe.

As funções de escala $\phi_{j,k}$ e as *Wavelets*, $\psi_{j,k}$, dizem-se ortogonais por respeitarem a condição expressa em (2.13):

$$\int_{-\infty}^{+\infty} \phi_{j,k}(x) \psi_{j,k}(x) dx = 0 \quad (2.13)$$

em que, $j \in \mathbb{Z}$ corresponde ao parâmetro de escala da função, e $k \in \mathbb{Z}$ corresponde à translação de $k/2^j$ em relação à função de escala e à *Wavelet* primária, dadas por $j = 0$ e $k = 0$. Tanto a função de escala como a *Wavelet* são definidas no conjunto dos reais ($\mathbb{R}$), através de escalonamento e translações das funções apresentadas.

O parâmetro translação corresponde à informação de tempo no domínio da transformada e o parâmetro escalonamento é o processo de compressão e dilatação do sinal [MALLAT 2009]. Na Figura 2.9 tem-se um exemplo de uma função de escala e uma *Wavelet* primária de Haar.

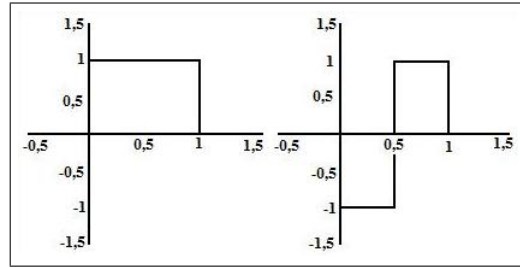


Figura 2.9: Função de escala e *Wavelet* primária de Haar.

Desta forma, pode-se dizer que a análise multiresolução através de *Wavelets* em tempo discreto corresponde a sucessivas passagens de filtros passa-faixa decompondo os sinais a cada passo em aproximação e detalhe.

A Figura 2.10 ilustra este esquema sendo aplicado a uma imagem de entrada, onde são passados filtros sucessivos nas linhas e nas colunas, gerando as novas escalas. O processo inverso, que realiza o somatório dos subespaços, pode recompor a imagem original. Utilizando as equações de dilatação, estas representam os filtros passa-baixa que geram as aproximações da imagem original. Já as equações de translação representam os filtros passa-alta e são responsáveis para obtenção dos detalhes da imagem original. A decomposição das funções de detalhes são as seguintes: detalhes na vertical (filtro passa-alta nas linhas e passa-baixa nas colunas), detalhes na horizontal: (filtro passa-baixa nas linhas e passa-alta nas colunas), detalhes na diagonal: (filtro passa-alta nas linhas e nas colunas).

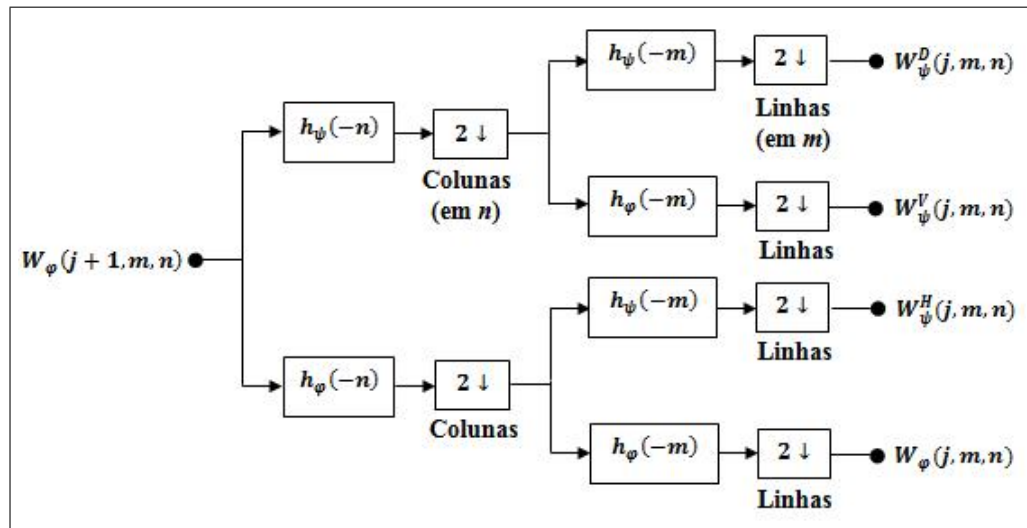


Figura 2.10: Decomposição da *wavelet*.

Então, utilizando uma imagem inicial com escala de $(j+1)$, sendo m linhas e n colunas, é mostrada na Figura 2.11 a decomposição desta imagem em dois níveis.

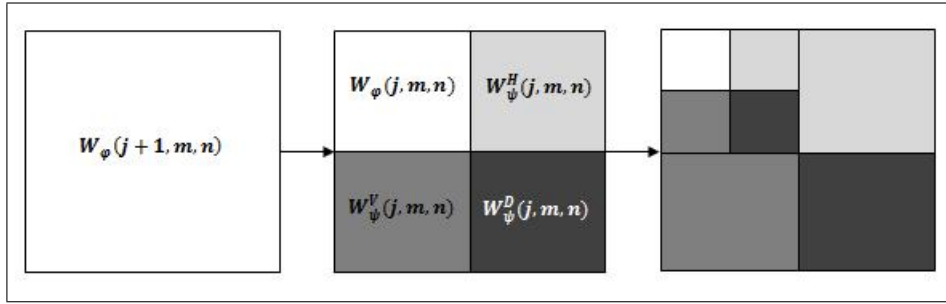


Figura 2.11: Representação multiresolução de uma imagem.

A imagem original e suas aproximações são as áreas mais claras da figura. As outras três subimagens restantes correspondem às três funções de detalhes da imagem original. No segundo passo, a imagem mais clara é novamente decomposta gerando uma nova imagem de aproximação e três novas subimagens de detalhe. Dessa forma, uma imagem de escala duas vezes menor foi gerada.

A Figura 2.12 mostra uma imagem inicial e o grau de refinamento desta obtido através da transformada *Wavelet*.

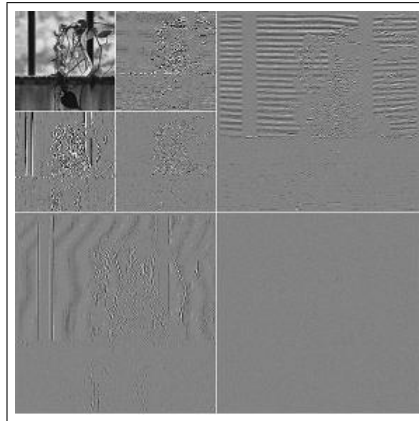


Figura 2.12: Análise multiresolução de uma imagem.

Esta forma de decompor e recompor imagens pode ser implementada com rapidez e eficácia, graças ao uso das transformadas *Wavelets*.

2.7.4 Compressão

O método de compressão de imagens baseado na transformada *Wavelet* deve tirar vantagem da esparsidade das imagens geradas ao longo da transformada. Um levantamento da função de densidade de probabilidade dos coeficientes da imagem transformada mostra que esta distribuição é tipicamente Laplaciana, isto é, existe uma grande quantidade de valores muito próximos de zero, que cai exponencialmente conforme se desloca para as extremidades do gráfico da distribuição.

Nota-se que a Transformada *Wavelet* discreta (DWT) pode ser escrita como uma

combinação linear de matrizes, ou seja, entre sua entrada e seus coeficientes. Por exemplo, assumindo 4 valores para o filtro h tem-se em (2.14) :

$$w_n = h_3x_{n-3} + h_2x_{n-2} + h_1x_{n-1} + h_0x_n \quad (2.14)$$

da mesma forma, obtém-se (2.15) :

$$w_n = [h_3 + h_2 + h_1 + h_0] \begin{bmatrix} x_{n-3} \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} \quad (2.15)$$

e ainda pode ser generalizada por (2.16) :

$$\begin{bmatrix} w_n \\ w_{n+1} \\ w_{n+2} \\ w_{n+3} \\ \vdots \\ w_{n+m} \end{bmatrix} = \begin{bmatrix} h_3 & h_2 & h_1 & h_0 & 0 & 0 \cdots & 0 \\ 0 & h_3 & h_2 & h_1 & h_0 & 0 \cdots & 0 \\ 0 & 0 & h_3 & h_2 & h_1 & h_0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & h_3 & h_2 & h_1 & h_0 \end{bmatrix} \begin{bmatrix} x_{n-3} \\ x_{n-2} \\ x_{n-1} \\ x_n \\ x_{n+3} \\ x_{n+2} \\ x_{n+1} \\ \vdots \\ x_{n+m} \end{bmatrix} \quad (2.16)$$

Observa-se que esta esparsidade possibilita grandes ganhos computacionais (redução do uso de memória espacial e temporal), e rapidez em esquemas de compressão de imagens. A Figura 2.13 mostra um exemplo de compressão utilizando a transformada *Wavelet*.



(a) Sem compressão. (b) Com compressão.

Figura 2.13: Representação da compressão de uma imagem.

2.8 Conclusão

Este capítulo teve como objetivo fornecer toda a fundamentação teórica relacionada ao processamento digital de imagens utilizado neste trabalho, conteúdo esse, essencial para um melhor entendimento do sistema proposto no capítulo 4 .

O custo computacional é extremamente importante quando se trabalha com processamento de imagens, ou seja, com os limitados recursos de *hardware* com os quais são realizadas as implementações. Entender a matemática envolvida na construção dos algoritmos, analisar o estado da arte de cada conteúdo em vez de apenas um todo, verificar a facilidade de implementação das propostas, realizar adaptações ou até utilizar novas abordagens para a resolução dos problemas, fazem parte de fatores determinantes para obtenção de bons resultados.

O capítulo mostrou: diferenças entre os sistemas intrusivos e não-intrusivos, vantagens e desvantagens de cada caso; definições sobre o espectro de cores e como essas informações preliminares podem influenciar no decorrer do sistema; a segmentação de imagens, verificando principalmente os procedimentos para a limiarização de imagens; a detecção da face, exibindo as principais vertentes de pesquisa sobre o assunto, inclusive sobre a SMQT, técnica utilizada neste trabalho; também foi visto como calcular o centro de massa de imagens e por fim a utilização de transformadas *Wavelets* para a extração de características e a análise multiresolução de imagens.

Este capítulo apresentou uma técnica de multiresolução, a transformada *Wavelet*, a fim de efetuar a filtragem, compressão e extração de características presentes nas imagens. Desta forma, pode-se criar um vetor de descritores que deve ser utilizado para treinamento e classificação dos padrões selecionados, utilizando redes neurais. O vetor de descritores contém elementos cujos valores descrevem apuradamente o conteúdo da imagem e ocupam menos espaço que a imagem representada *pixel* a *pixel*, de tal sorte a otimizar o sistema como um todo [PEIXOTO 2009].

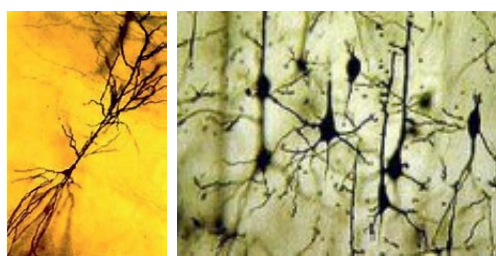
O entendimento deste conjunto de técnicas apresentadas para utilização no processamento digital de imagens irá melhor possibilitar uma maior motivação e melhor compreensão de seu uso em capítulos posteriores.

Capítulo 3

Redes Neurais Artificiais

As redes neurais artificiais, ou apenas redes neurais, representam uma tecnologia que possui raízes em muitas disciplinas tais como: neurociência, matemática, estatística, física, ciências da computação e engenharia. São desenvolvidas aplicações nas mais variadas áreas, como: modelagem, análise de séries temporais, reconhecimento de padrões, processamento de sinais e controle. Tudo isso é possível devido às propriedades típicas das redes neurais como, por exemplo, a habilidade de aprender a partir de dados de entrada com ou sem professor [HAYKIN 2001]. As redes neurais podem ser definidas como conjuntos de estruturas de dados e algoritmos que têm suas inspirações em ciências como a biologia, criadas para reter, comparar ou gerar dados relacionados a um sistema. Conforme [HAYKIN 2001], as redes neurais artificiais tiveram seu início com o trabalho de dois cientistas, MacCulloch, um psiquiatra e neuroanatomista e Pitts, um matemático.

Os dois definiriam o que provavelmente seria uma representação matemática de um neurônio e como funcionaria de forma rudimentar o processamento das informações a partir destes neurônios. Um neurônio biológico e uma rede neural biológica podem ser vistos na Figura 3.1.



(a) Neurônio. (b) Uma rede neural.

Figura 3.1: Neurônios biológicos.

A Figura 3.2 mostra um modelo não linear de um neurônio artificial. Os dados de entrada do neurônio artificial são representados pelas variáveis x_i , os pesos sinápticos são representados por w_{ki} , em que o índice i refere-se à entrada i e o índice k ao neurônio k ; b_k representa o valor do bias; $\phi(\cdot)$ a função de ativação do neurônio e, por fim, y_k representa a saída do neurônio.

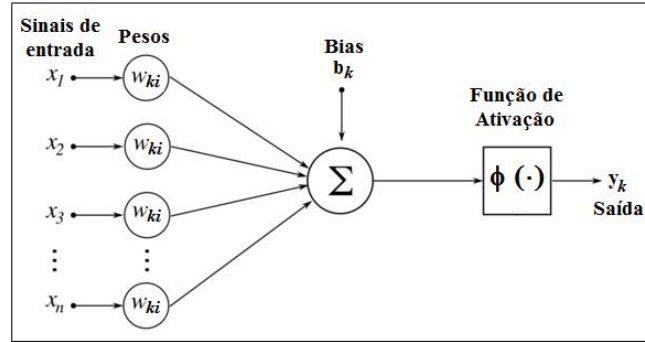


Figura 3.2: Neurônio artificial.

Os dados de entrada são organizados em um vetor de entrada dado por $x_i = [x_1, \dots, x_m]^T$, geralmente com $x_i \in \mathfrak{R}$, formando uma representação do estado ou da informação a ser processada pela rede neural. O vetor de entrada é ponderado pelos respectivos pesos sinápticos w_{ki} e então acumulado pelo combinador central, representado pela função aditiva, juntamente a um bias que tem a função de aumentar ou reduzir o valor do acumulador. A expressão (3.1) representa o valor do acumulador:

$$v_k = b_k + \sum_{i=1}^m x_i w_{ki} \quad (3.1)$$

O valor do acumulador v_k , também conhecido como campo local induzido ou ainda potencial de ativação, é utilizado pela função de ativação $\phi(\cdot)$ para gerar a saída y_k do neurônio:

$$y_k = \phi(v_k) \quad (3.2)$$

Geralmente, a função de ativação $\phi(\cdot)$ é do tipo não linear. Dentre as mais comuns estão a função de limiar, a função linear por partes e a função sigmóide. As funções sigmóides são as mais frequentemente utilizadas no desenvolvimento das redes neurais.

Existe uma variação nas arquiteturas de algoritmos que fazem uso da estrutura do neurônio artificial, como por exemplo, o perceptron de uma única camada, as redes compostas por múltiplos perceptrons que fazem uso de várias camadas de neurônios artificiais e as redes de base radial, que fazem uso de uma variação do neurônio artificial. Neste trabalho, são utilizadas as redes perceptrons de múltiplas camadas, cujas características serão detalhadas na seção 3.2.

3.1 Processos de aprendizagem

Uma característica importante em uma rede neural é a sua habilidade de aprender a partir do ambiente em que está inserida e com isso aprimorar seus resultados.

Em sua aprendizagem, uma rede neural faz uso de um processo iterativo de ajustes aplicados a seus pesos sinápticos e níveis de bias. A cada nova interação, sua aprendizagem sobre o ambiente se torna mais precisa. Existe uma definição adaptada por Mendel e McClaren [MCLAREN 1970] que diz o seguinte: “Aprendizagem é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira como as modificações dos parâmetros ocorrem”.

O processo de aprendizagem de uma rede neural pode ser realizado de duas formas: o aprendizado com professor ou supervisionado e o aprendizado não-supervisionado, sendo este realizado apenas através da experiência adquirida pela rede. No aprendizado supervisionado, a rede neural é treinada com pares entrada-saída desejados. Desta forma, fazendo ajustes nos pesos sinápticos, é possível minimizar a diferença entre a saída apresentada pela rede e a saída desejada, sendo realizadas tantas iterações quantas forem necessárias para se chegar ao erro desejado. Na aprendizagem não-supervisionada, a rede não possui informações sobre a saída desejada. Isso faz com que a rede neural seja treinada apenas com valores de entrada, objetivando organizar sua estrutura de forma a reconhecer padrões, detectar características comuns aos dados e fazer uso de processos de competição e cooperação para, com isso, realizar uma auto-organização.

Um conjunto bem definido de regras para a solução de um dado problema de aprendizagem é denominado algoritmo de aprendizagem. O mesmo possui algumas variedades, basicamente diferenciando entre si a partir da forma como ocorre o ajuste dos pesos sinápticos dos neurônios. Algumas formas importantes de aprendizagem podem ser destacadas: correção de erro (regra delta), competitiva (regra de Kohonen), baseada em memória, Hebbiana e aprendizagem de Boltzmann. Neste trabalho, são utilizadas as técnicas de aprendizagem supervisionadas, mais especificamente, a aprendizagem por correção de erro.

3.2 Rede perceptrons de múltiplas camadas (MLP)

Redes do tipo Perceptrons de Múltiplas Camadas, também conhecidas como MLP (do inglês *Multilayer Perceptrons*) constituem uma das arquiteturas mais difundidas das Redes Neurais Artificiais. Uma rede MLP pode ser observada a partir da Figura 3.3 e pode ser assim descrita: Uma camada de entrada, que possui nós de entrada na mesma quantidade dos sinais de entrada, uma ou mais camadas ocultas de neurônios e uma camada de saída com um número de neurônios igual ao número de sinais de saída. O sinal de entrada se propaga para frente através das camadas, primeiro pela camada de entrada, depois pelas camadas ocultas, tantas quantas forem utilizadas, até chegar à camada de saída. Esta arquitetura caracteriza uma rede alimentada diretamente com múltiplas camadas. Em cada passagem por entre as camadas, a entrada é ponderada por um peso sináptico e acumulada juntamente ao bias, formando um campo local induzido, que é então utilizado pela função de ativação em cada neurônio das várias camadas da rede.

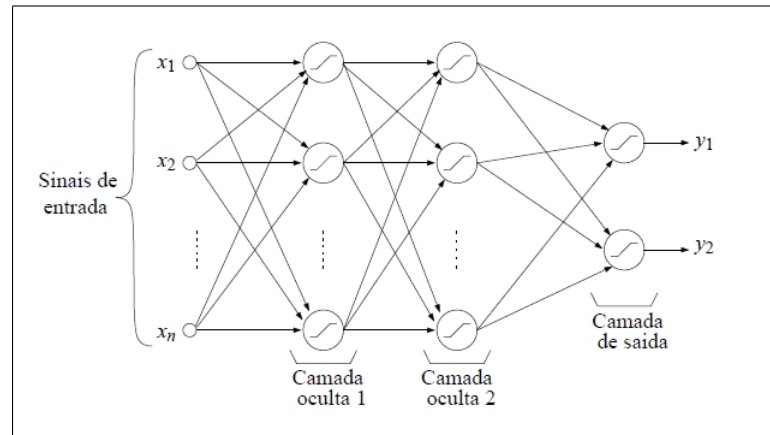


Figura 3.3: Arquitetura genérica de uma rede neural artificial do tipo MLP com duas camadas ocultas.

As redes neurais MLP costumam interagir com o ambiente, durante sua fase de treinamento, da maneira como é mostrada na Figura 3.4. O ambiente fornece à rede um valor de entrada que o descreve; este valor é utilizado pela rede e pelo professor, por isso, a aprendizagem é dita supervisionada. A rede fornece então uma saída, denominada saída real, e o professor gera uma saída que seria a resposta desejada, ou resposta ideal, dada a entrada fornecida. As duas saídas são combinadas de forma a gerar um valor de diferença entre as duas. Este valor, denominado sinal de erro, é utilizado então pelo algoritmo de treinamento da rede para reajustar seus parâmetros de forma a convergir para o resultado ideal.

As redes MLP fazem uso de um algoritmo comumente utilizado e já bem definido na literatura, conhecido como algoritmo de retropropagação do erro. O algoritmo da retropropagação consiste em ciclos de treinamentos compostos por: apresentações aleatórias de vetores de entradas na RNA, obtenção da diferença entre a resposta fornecida pela rede e a resposta desejada, utilização dessa diferença como o argumento de uma alteração realizada sob o conjunto dos pesos sinápticos da rede, sendo esta alteração baseada no gradiente descendente. Os ciclos de treinamento podem fazer a atualização dos pesos sinápticos de forma interativa (*online*), ou seja, a cada passo do algoritmo, ou em forma de lote (*batch* ou *offline*), após determinada quantidade de passos.

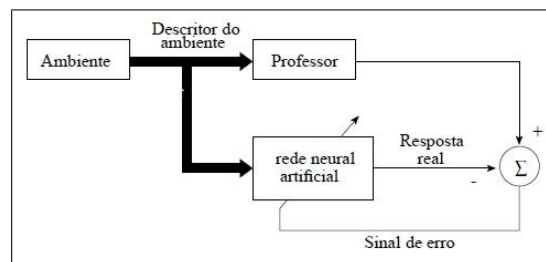


Figura 3.4: Aprendizado supervisionado.

3.3 Algoritmos de treinamento

Um conjunto pré-estabelecido de regras bem definidas para a solução de um problema de aprendizagem é denominado algoritmo de treinamento. Não existe um algoritmo de aprendizagem genérico para o projeto de redes neurais. Em vez disso, tem-se uma variedade de algoritmos de treinamento, cada um oferecendo vantagens específicas para cada modelo utilizado. O processo de treinamento requer um conjunto de exemplos que refletem o comportamento da própria rede a elementos de entrada e saída. Durante o treinamento, os pesos e *biases* são ajustados iterativamente, com o intuito de minimizar a função custo ou erro, mais comumente representada pelo erro médio quadrático (*Mean Square Error* - MSE), entre a saída da rede e a saída desejada, ou seja, fornecida à rede. Todos os algoritmos para esse tipo de rede consistem em obter o gradiente da função de desempenho da rede, a fim de ajustar os pesos sinápticos, minimizando a função erro/custo. Em geral, o gradiente é determinado utilizando-se a técnica denominada retropropagação (*backpropagation*), uma generalização da regra de aprendizado Widrow-Hoff, aplicada a redes neurais de múltiplas camadas que trabalham com funções de transferência não-lineares diferenciáveis [BEALE 1996].

Porém, para a solução de vários problemas práticos, o algoritmo do gradiente descendente tradicional não é o mais adequado, então variações do algoritmo de retropropagação são utilizadas, baseadas em outras técnicas de otimização, considerados algoritmos de alto desempenho, que podem convergir até 100 vezes mais rápido que o algoritmo de retropropagação tradicional [HAGAN 2009]. São alguns exemplos destes algoritmos, o *Resilient Backpropagation* (RPROP), o método do gradiente conjugado e o método de Newton. Estes algoritmos mais robustos são divididos em duas categorias. A primeira utiliza técnicas heurísticas desenvolvidas a partir de algoritmos de retropropagação tradicionais (método do gradiente descendente), como é o caso do RPROP. Os algoritmos que se enquadram nesta categoria são também chamados de métodos de primeira ordem. A segunda categoria faz uso de técnicas numéricas de otimização, tais como os algoritmos: Levenberg-Marquardt, BFGS Quasi-Newton, Gradiente Conjugado Escalar (*Scaled Conjugate Gradient*) e Gradiente Conjugado com Reinicialização Powell-Beale (*Conjugate Gradient with Powell-Beale Restarts*) [HAGAN 2009]. Estes algoritmos são também conhecidos como métodos de segunda ordem.

3.3.1 Algoritmo de retropropagação (*backpropagation*)

A implementação mais simples da técnica de retropropagação é responsável pela atualização dos pesos e *biases* na direção em que a função de desempenho da rede decresce mais rapidamente (sentido negativo do gradiente). Uma iteração deste algoritmo pode ser expressa por:

$$w_{i+1} = w_i + \delta g_i \quad (3.3)$$

em que w_i é o vetor atual de pesos e *biases*, g_i é o gradiente correspondente ao vetor w_i e δ_i é a taxa de aprendizado.

Existem duas formas de implementação do método do gradiente descendente: o treinamento por lote ou *batch*, e o treinamento incremental. No modo incremental, o gradiente é implementado de forma que os pesos sejam atualizados após cada elemento de entrada ser apresentado à rede. Já no modo *batch*, todas as entradas são aplicadas à rede antes da atualização dos pesos. O gradiente calculado para cada exemplo de treinamento fornecido à rede é adicionado aos demais para então ser feita a atualização de pesos e *biases* [BEALE 1996]. A taxa de aprendizado δ_i é multiplicada inúmeras vezes pela parte negativa do gradiente, de forma que, quanto maior seu valor, maior o passo de atualização. Assim, se a taxa de aprendizado for muito elevada, o algoritmo pode apresentar oscilações, tornando-se instável; por outro lado, se a taxa de aprendizado é muito baixa, o algoritmo torna-se lento, demorando muito para convergir [BEALE 1996]. Deste modo, para melhorar o desempenho do algoritmo do gradiente descendente, foi proposta uma taxa de aprendizado adaptativa, responsável por tornar o aprendizado mais rápido e estável. Este novo método foi denominado *resilient backpropagation*.

3.3.2 Algoritmo *resilient backpropagation* (RPROP)

O RPROP é um eficiente esquema de aprendizagem que executa a adaptação direta da atualização dos pesos sinápticos baseada na informação do gradiente local. Uma diferença crucial entre o RPROP e o algoritmo *Backpropagation* é o fato de que o esforço da adaptação dos pesos não é prejudicado pelo comportamento do gradiente, sendo introduzido um valor de atualização δ_{ij} para cada peso sináptico, responsável por determinar apenas a amplitude da atualização do peso. Essa atualização adaptativa evolui durante o processo de treinamento, baseada na visão local da função custo E , de acordo com a seguinte regra de aprendizagem [BRAUN 1993]:

$$\delta_{ij}^{(t)} = \begin{cases} \eta^+ \cdot \delta_{ij}^{(t-1)}, & \text{se } \frac{\partial E^{(t-1)}}{\partial w^{ij}} \cdot \frac{\partial E^{(t)}}{\partial w^{ij}} > 0 \\ \eta^- \cdot \delta_{ij}^{(t-1)}, & \text{se } \frac{\partial E^{(t-1)}}{\partial w^{ij}} \cdot \frac{\partial E^{(t)}}{\partial w^{ij}} < 0 \\ \delta_{ij}^{(t-1)}, & \text{se } \frac{\partial E^{(t-1)}}{\partial w^{ij}} \cdot \frac{\partial E^{(t)}}{\partial w^{ij}} = 0 \end{cases} \quad (3.4)$$

em que $0 < \eta^- < 1 < \eta^+$ e t representa o número de épocas do treinamento. Seguindo a regra mostrada em (3.4), toda vez que a derivada parcial correspondente ao peso w_{ij} muda seu sinal em relação ao passo de tempo anterior, significa que a última atualização (ajuste) foi muito alta e o algoritmo passou por um mínimo local. Assim, o valor de atualização δ_{ij} decresce pelo fator η . Caso a derivada mantenha o mesmo sinal, o valor de atualização é incrementado de forma a acelerar a convergência. Em resumo, se a derivada é positiva (aumentando o erro), o peso é reduzido pelo seu

valor de atualização; caso a derivada seja negativa, o valor de atualização passa a ser então positivo [BRAUN 1993]:

$$\delta_{ij}^{(t)} = \begin{cases} -\delta_{ij}^{(t)}, & \text{se } \frac{\partial E^{(t)}}{\partial w^{ij}} > 0 \\ +\delta_{ij}^{(t)}, & \text{se } \frac{\partial E^{(t)}}{\partial w^{ij}} < 0 \\ 0, & \text{caso contrário} \end{cases} \quad (3.5)$$

No entanto, existe uma exceção: se a derivada parcial trocar de sinal, isto é, se o passo anterior for grande demais e o mínimo for ultrapassado, a atualização do peso δw_{ij} é revertida:

$$\delta_{ij}^{(t)} = \delta_{ij}^{(t-1)}, \quad \text{se } \frac{\partial E^{(t-1)}}{\partial w^{ij}} \cdot \frac{\partial E^{(t)}}{\partial w^{ij}} > 0 \quad (3.6)$$

Por causa disto, a derivada deve mudar seu sinal novamente no passo seguinte, para evitar que ocorra uma nova punição do valor de atualização. Não deve haver adaptação do valor de atualização no passo posterior. Uma forma prática de evitar isso é fazendo com que $\frac{\partial E^{(t-1)}}{\partial w^{ij}} = 0$. Os valores de atualização e os pesos somente são modificados depois que todo o conjunto de treinamento é apresentado à rede, o que caracteriza aprendizagem por lote ou *batch* [BRAUN 1993]. Inicialmente, todos os valores de ajuste são iguais à constante δ_0 , que é um dos parâmetros do RPROP. Uma vez que a constante δ_0 determina diretamente a amplitude do primeiro ajuste dos pesos, ela pode ser escolhida de acordo com a magnitude dos pesos iniciais, por exemplo, $\delta_0 = 0,1$ [BRAUN 1993]. A escolha desse valor não é crítica, já que seu valor é adaptado enquanto o treinamento ocorre. No treinamento da rede através do algoritmo RPROP, para evitar uma variação excessiva dos pesos, define-se um parâmetro para o valor máximo de ajuste, δ_{max} , que assume um valor igual a 50, sugerido por Riedmiller e Braun (1993). Os fatores de acréscimo e decréscimo são fixados em $\eta^+ = 1,2$ e $\eta^- = 0,5$, respectivamente. Estes valores são baseados em considerações teóricas e empíricas. Sendo assim, o número de parâmetros fica reduzido a dois, δ_0 e δ_{max} [BRAUN 1993].

3.3.3 Algoritmos de gradiente conjugado

Este tipo de algoritmo caracteriza-se por ser implementado ao longo da direção do gradiente conjugado, o que proporciona uma convergência mais rápida que a obtida com o método do gradiente descendente [BEALE 1996]. Na maioria dos algoritmos de gradiente conjugado, a taxa de aprendizado é ajustada a cada iteração. Uma busca pela solução ótima é realizada na direção do gradiente conjugado, fazendo com que a função de desempenho da rede seja minimizada. Dois exemplos destes algoritmos foram analisados e são descritos sucintamente:

a) Gradiente Conjugado com Reinicialização *Powell-Beale*

Em todos os algoritmos que utilizam a técnica do gradiente conjugado, a direção de busca é reiniciada periodicamente no sentido negativo do gradiente. O critério de parada desta atualização ocorre quando o número de iterações torna-se igual ao número de parâmetros da rede (pesos e *biases*). Contudo, existem outros meios que proporcionam a busca pela solução ótima concomitantemente à eficiência do treinamento. Um exemplo é o método de busca proposto por Powell (1982) e Beale (1972). A condição para o reinício da busca é a falta de ortogonalidade entre o gradiente atual e o anterior, o que pode ser testado de acordo com a desigualdade expressa em (3.7). Se esta condição é satisfeita, a busca pela solução ótima é reiniciada em direção à parte negativa do gradiente.

$$|g_{i-1}^T g_i| \geq 0.2 \|g_i\|^2 \quad (3.7)$$

b) Gradiente Conjugado Escalar (*Scaled Conjugate Gradient - SCG*)

O espaço de busca requerido pela maioria dos algoritmos de gradiente conjugado demanda um custo computacional elevado, uma vez que a resposta da rede é computada várias vezes para cada elemento de entrada. O algoritmo do gradiente conjugado escalar [MOLLER 1993] foi desenvolvido para diminuir o custo computacional na obtenção do espaço de busca. Este algoritmo combina a eficiência da técnica do gradiente conjugado com o algoritmo *Levenberg-Marquardt* [BEALE 1996].

3.3.4 Algoritmos baseados no método Quasi-Newton

O método de Newton representa uma alternativa para os métodos de gradiente estudados. O método tem como característica uma maior rapidez na convergência dos resultados (em relação aos métodos de gradiente conjugado), contudo, consiste em uma técnica complexa e que requer elevado custo computacional, especialmente na implementação da matriz Hessiana, primeiro passo na implementação em redes neurais alimentadas adiante. Entretanto, existe uma classe de algoritmos baseados no método de Newton que não precisam calcular derivadas de segunda ordem; são os chamados métodos de otimização Quasi-Newton. Estes métodos trabalham com uma matriz Hessiana aproximada, que é atualizada a cada iteração do algoritmo, em função do gradiente. O método que emprega tais características e que tem sido mais usado é o desenvolvido por Broyden, Fletcher, Goldfarb e Shanno, denominado, portanto, método BFGS [SCHNABEL 1983]. Este algoritmo necessita de um maior custo computacional, em termos de processamento e memória, que os métodos de gradiente conjugado, embora sempre convirja com poucas iterações. A matriz Hessiana aproximada deve ser armazenada e sua dimensão é do tipo $n \times n$, onde n representa o número de pesos e *biases* utilizados na rede. Para redes neurais cujo conjunto de treinamento seja muito extenso, é mais adequado o uso do algoritmo RPROP ou até mesmo de uma das técnicas de gradiente conjugado.

3.3.5 Algoritmo Levenberg-Marquardt (LM)

Assim como os algoritmos que usam o método Quasi-Newton, o algoritmo Levenberg-Marquardt [MENHAJ 1999] foi desenvolvido para aumentar a velocidade de treinamento de algoritmos de segunda ordem, sem, no entanto, implementar a matriz Hessiana. Sempre que a função de desempenho da rede possa ser analisada pela soma de seus quadrados (forma utilizada em redes neurais alimentadas adiante), a matriz Hessiana pode ser aproximada para:

$$H = J^t J \quad (3.8)$$

e o gradiente pode ser implementado de acordo com (3.9):

$$g = J^t e \quad (3.9)$$

em que J é a matriz Jacobiana, que armazena as derivadas de primeira ordem da função de desempenho da rede, com relação aos pesos e *biases*, e a variável e é o vetor responsável por armazenar os erros apresentados pela rede. A matriz Jacobiana pode ser implementada através da técnica de retropropagação tradicional, o que representa muito menos complexidade se comparada à implementação da matriz Hessiana [HAGAN 2009].

O algoritmo Levenberg-Marquardt utiliza a matriz Jacobiana no lugar da matriz Hessiana, na atualização dos pesos, como mostrado em (3.10):

$$x_{i+1} = x_i - [J^t J - \mu I]^{-1} J^t e \quad (3.10)$$

De acordo com (3.10), quando a variável escalar μ for igual a zero, o método assemelha-se ao método de Newton, na aproximação da matriz Hessiana; por outro lado, quando o valor de μ é alto, o método comporta-se como o gradiente descendente com um pequeno passo de atualização. Como o método de Newton é mais rápido e mais preciso na detecção de erros, o objetivo do algoritmo Levenberg-Marquardt é, portanto, tornar-se tão eficiente quanto ele. Deste modo, força-se a variável μ a decrescer após cada iteração bem sucedida. Esta variável só terá seu valor acrescido quando uma determinada atualização for responsável por maximizar (e não minimizar) a função de desempenho da rede. Isto possibilita que a função custo seja sempre reduzida em cada iteração do algoritmo [HAGAN 2009].

3.4 Conclusão

Aplicações envolvendo processamento de sinais e redes neurais são bem comuns nos dias de hoje. No contexto desse trabalho, os sinais analisados são as imagens adquiridas a partir de uma *webcam*, conforme visto no capítulo 2.

A utilização das redes neurais é de fundamental importância neste trabalho, pois sua utilização está vinculada à obtenção dos resultados de classificação da direção do olhar.

Algumas etapas fundamentais para a utilização das redes neurais são: arquitetura, coleta de dados, configuração de parâmetros, treinamento e generalização dos resultados. Cada uma das etapas possui especial importância e por isso foram trabalhadas uma a uma neste trabalho. Seus detalhes estão apresentados no capítulo 4, onde ocorre de fato a sua implementação.

Este capítulo teve como objetivo fundamentar as redes neurais artificiais, incluindo informações específicas e direcionadas ao trabalho desenvolvido. Foram apresentados os princípios de funcionamento das redes neurais, os principais processos de aprendizagem e os algoritmos de treinamento testados e utilizados neste trabalho.

Capítulo 4

Sistema Não Intrusivo para Estimação da Direção do Olhar

O sistema proposto neste trabalho estabelece uma organização lógica de todas as etapas envolvidas no processo de estimação da direção do olhar. A Figura 4.1 permite visualizar um diagrama de blocos da composição e funcionamento do sistema. Cada bloco deste diagrama representa uma etapa fundamental no desenvolvimento do trabalho, que vai desde a aquisição das imagens a partir de uma simples *webcam*, até a estimação da direção do olhar utilizando um classificador neural.

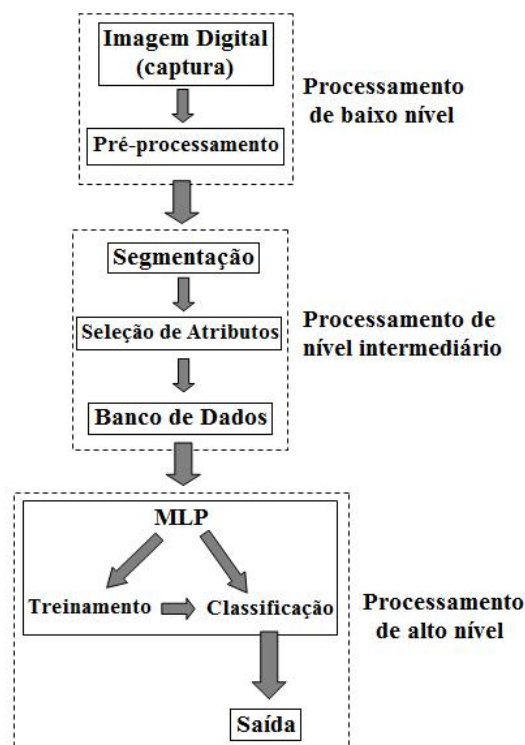


Figura 4.1: Diagrama de blocos do sistema desenvolvido.

Utilizando os fundamentos apresentados nos capítulos anteriores, torna-se possível montar uma estrutura que permita entender todo o fluxo de informação na sua ordem adequada à solução de problemas locais e ao objetivo desejado. Além disso, é possível perceber os motivos que levaram os autores a utilizar ou desenvolver tais soluções, algoritmos ou técnicas apresentadas.

Observa-se pela Figura 4.1 que o diagrama de blocos possui três partes principais: o processamento de baixo nível, contendo os blocos de captura e pré-processamento da informação; o processamento de nível intermediário, envolvendo os blocos de segmentação das imagens, a seleção de atributos e a construção de um banco de dados e, por fim, o processamento de alto nível, fazendo a interpretação e o reconhecimento dos padrões com os blocos de treinamento e a classificação neural. As seções seguintes permitem o detalhamento deste diagrama.

4.1 Processamento de baixo nível

Esta é a primeira etapa a ser vencida. O primeiro passo em busca do objetivo deste trabalho, a estimação da direção do olhar. Tudo começa com a aquisição das imagens e o pré-processamento das mesmas.

4.1.1 Capturando e pré-processando imagens

Inicialmente, a imagem é capturada a partir de uma simples *webcam*, modelo *Logitech QuickCam Communicate STX*, resolução máxima de 1,3 *Mpixels* - baseada em padrões da FCC (*Federal Communications Commission*) que está conectada a um *notebook* com as seguintes configurações: PC Intel Celeron M, 1,6 GHz, 2 GB RAM, Monitor de 15,4" e Sistema Operacional Linux Ubuntu 9.10. Foi utilizado o *software* comercial Matlab 2009 versão estudante em todo o desenvolvimento do sistema.

Neste trabalho, para a captura das imagens, utilizou-se o conceito de sistemas não-intrusivos, cuja pretensão é facilitar o manuseio dos equipamentos envolvidos e assim construir um sistema de baixo custo aquisitivo e fácil manuseio. A Figura 4.2 exibe o cenário utilizado. Observa-se que o usuário está em uma posição ergonomicamente correta e que a distância entre os olhos e a tela do computador é de aproximadamente 60 cm. Pequenas variações na distância (até 6 cm) entre usuário e tela do computador, além de leves variações na luminosidade (até 10%) do ambiente foram consideradas para a aquisição das imagens.

Depois da imagem ser capturada a partir de uma simples *webcam*, é realizado um pré-processamento, cujo objetivo é melhorar a imagem e assim facilitar os processos seguintes. Neste ponto foram aplicadas técnicas para o realce de contrastes e controle de luminosidade do ambiente.

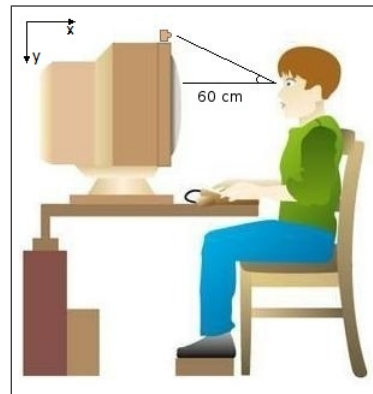


Figura 4.2: Cenário utilizado para a captura das imagens.

4.2 Processamento de nível intermediário

O processamento de nível intermediário consiste na segunda etapa do sistema, onde quase todos os conceitos sobre processamento de imagens, abordados no capítulo 2, serão aplicados, objetivando sempre uma boa relação entre resultado e desempenho.

4.2.1 Segmentação das imagens

De forma geral, a segmentação de imagens consiste em dividir uma imagem de entrada em partes ou objetos constituintes. Esta é considerada uma tarefa difícil, mas que pode favorecer enormemente a solução dos problemas de imageamento.

Delimitação da área da face: Neste trabalho, tal delimitação tem como objetivo eliminar as informações desnecessárias acrescentadas pelo ambiente. Assim sendo, a detecção da face realizada neste trabalho baseou-se em uma técnica de processamento de imagens conhecida como SMQT (*Successive Mean Quantization Transform*) [CLAESSON 2005, CLAESSON 2007].

As variações ambientais são muito comuns, porém as principais características do rosto permanecem constantes. Isto proporciona melhores condições para a extração de objetos, por exemplo, olhos, nariz e boca. Vale ressaltar que no processo de aquisição das imagens em um ambiente, existe grande quantidade de informações desnecessárias (tais como os dados que estão localizados fora da área do rosto) e estes são fatores que podem prejudicar a precisão dos resultados durante o processamento da imagem. A Figura 4.3 mostra o resultado de uma detecção da face realizada neste trabalho. Observa-se que a imagem (do ambiente) obtida a partir de uma *webcam*, é utilizada para detectar a face, resultando nesta área selecionada pelo retângulo em vermelho da Figura 4.3.

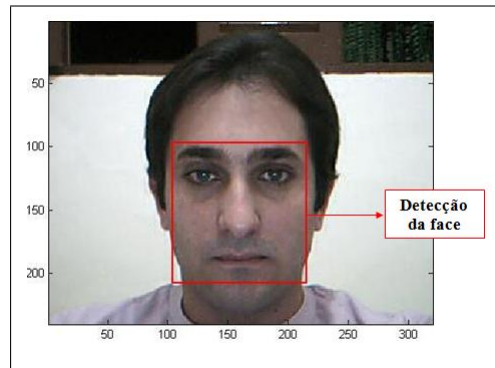


Figura 4.3: Detecção da face.

Detecção dos olhos: Depois da detecção da face, verifica-se que muita informação desnecessária foi eliminada da imagem, mas ainda é preciso continuar até que se chegue à detecção dos olhos, mais especificamente de um dos olhos, já que neste caso seria redundância utilizar os dois olhos.

Alguns bons métodos para detecção ou delimitação dos olhos podem ser vistos, por exemplo, em [DEL 2007], onde a detecção dos olhos é realizada com base na análise da textura dos *pixels* a fim de detectar alguns padrões. Em [YEPENG 2007], o método para localizar os olhos das imagens faciais é apresentado com base nas multi-informações faciais. Outro método interessante é proposto em [HAMID 2008], onde é utilizado o espaço de cores especiais YCbCr, e seus componentes fornecem informações úteis sobre os olhos.

Neste trabalho, foi desenvolvida uma técnica específica para a detecção do olho. Construiu-se um algoritmo de baixo custo computacional que envolve três etapas fundamentais. A primeira etapa é baseada em parâmetros da geometria facial, uma região dentro da área da face, previamente selecionada a fim de representar o local que deverá conter um dos olhos e sua respectiva sobrancelha. A Figura 4.4 mostra o início do processo.

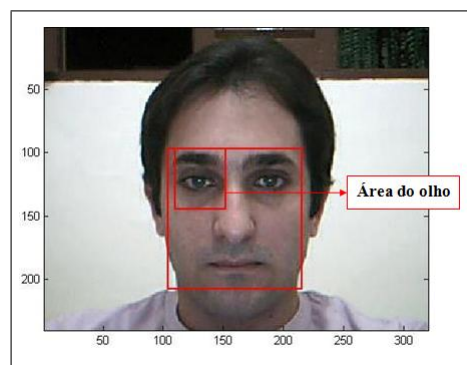


Figura 4.4: Delimitando a área do olho.

Na sequência, com a área do olho já delimitada, inicia-se a segunda parte, que consiste em continuar a segmentação, até que se chegue à detecção do olho, conforme

mostra a Figura 4.5. Vale salientar que esta segmentação é feita em uma região bem menor do que aquele inicial, onde a imagem foi capturada.

Utilizando os conceitos do processamento digital de imagens, visto no capítulo 2, são aplicados os seguintes algoritmos, na ordem: conversão para escala de cinza, binarização de Otsu e o algoritmo para detecção de objetos desenvolvido por [LINDA 1992], que torna possível identificar todos os objetos dentro de uma área selecionada, indicada pela Figura 4.5

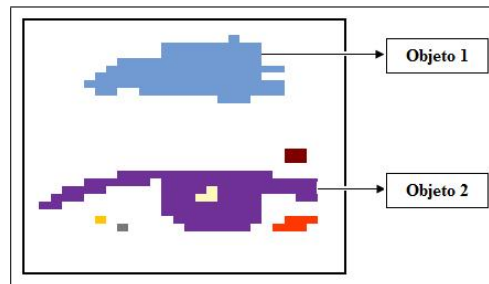


Figura 4.5: Identificação de objetos dentro da área do olho.

Por fim, a terceira etapa consiste em selecionar o objeto de interesse (o olho), conforme a Figura 4.6.

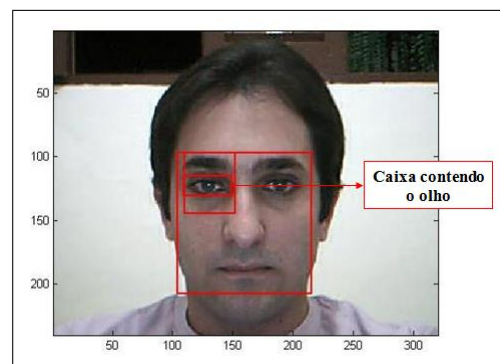


Figura 4.6: Detecção do olho.

Após a seleção do olho, deve-se localizar seu centro de massa, centralizando em uma caixa, na Figura 4.7. Esta caixa contém, portanto, as informações que serão enviadas ao banco de dados de utilização das redes neurais.



Figura 4.7: Identificação do centro de massa do olho.

4.2.2 Construindo o banco de dados

Entende-se por banco de dados (ou base de dados) um conjunto de registros dispostos em estrutura regular, que possibilita a reorganização e o uso das informações. Trabalhar com aquisição e manipulação de imagens, na maioria dos casos, significa estar diante de um grande volume de dados, daí a necessidade de se utilizar um banco de dados.

Ainda não é possível encontrar bancos de dados especializados para uso em sistemas não-intrusivos, e os disponíveis possuem limitações no sentido de não condizer com as características únicas de cada sistema, por exemplo, cada resolução e orientação das imagens.

Foi construído um banco de dados especializado para este trabalho. Este consiste em imagens de um dos olhos de usuários, em até 4 posições diferentes: olhar para a direita, olhar para a esquerda, olhar para cima e olhar para baixo.

A Figura 4.8 ilustra bem todo o processo de armazenamento das informações para o exemplo de uma imagem. Primeiramente, na fase do processamento de baixo nível, a imagem é capturada. Depois disso, no processamento de nível intermediário, são realizados diversos processos de segmentação de imagens, incluindo a detecção da face e do olho que, por sua vez, irão alimentar o banco de dados, utilizando ou não alguma seleção de atributos.

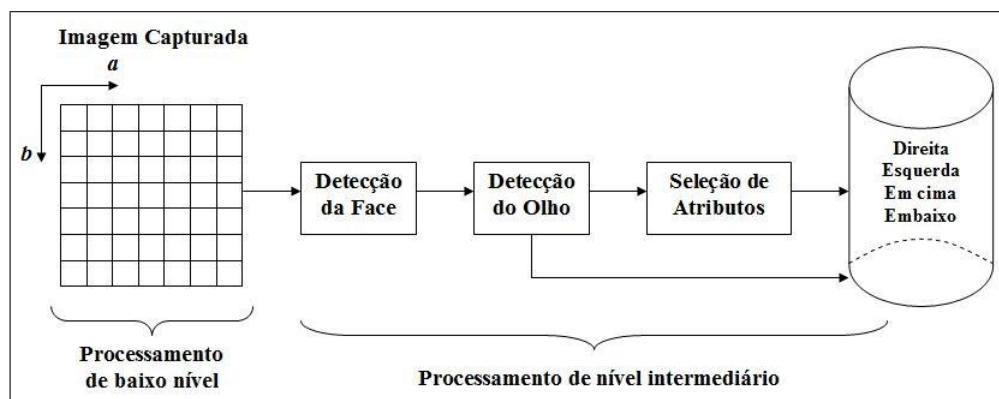


Figura 4.8: Processo de construção do banco de dados.

A imagem inicial de $(a \times b)$ pixels capturada, possui em seu tamanho, um fator determinante ligado ao custo computacional, podendo exigir grandes quantidades de memória espacial e temporal para seu processamento. Este mesmo tamanho traduz neste caso, um aumento da qualidade da imagem quando proporcional ao aumento do número de pixels. Desta forma, exige-se que se encontre o limite entre qualidade e tamanho desta imagem, pois assim, será possível conseguir um ganho computacional essencial em etapas de processamento a posteriori. Com o decorrer do processo, pode-se chegar a um vetor de $(a \times b)$ pixels, mantendo o conteúdo essencial das imagens, ou ainda extrair atributos importantes para o posterior reconhecimento de padrões. Sabe-se ainda que a redução adequada da matriz de dados, implicará diretamente na redução do custo computacional ligado ao treinamento da rede neural.

4.3 Processamento de alto nível

O processamento de alto nível trata da interpretação e reconhecimento das informações utilizando os conceitos apresentados no capítulo 3 sobre as redes neurais artificiais.

Modelos neurais, também conhecidos como sistemas *conexionistas*, fazem uso de componentes simples e interativos (os neurônios artificiais) que, de forma paralela e distribuída, podem aprender ou adaptar-se, desde que ajustem adequadamente essas conexões [LUGER 2004].

A Figura 4.9 mostra os detalhes do processamento de alto nível a partir do sistema desenvolvido. Inicialmente as imagens selecionadas passam por uma análise *Wavelet* (*Daubechies db3*) onde são selecionados os coeficientes necessários para construção do conjunto de treinamento e classificação da rede neural.

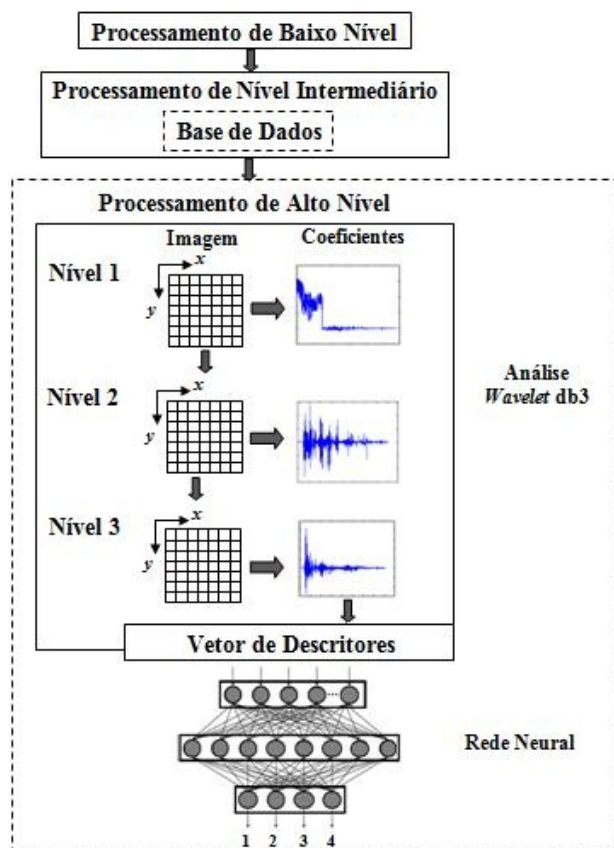


Figura 4.9: Processamento de alto nível.

4.3.1 Redes neurais: treino e classificação

Para a implementação do sistema desenvolvido, exibido na Figura 4.9, utilizou-se na arquitetura da rede, uma classe chamada de Redes Alimentadas Diretamente

com Múltiplas Camadas, que se distinguem de outras classes devido à presença de uma ou mais camadas ocultas.

Aprendizagem: Com uma arquitetura já selecionada para a rede neural, tem-se uma camada de entrada consistindo de nós de fontes iguais em número de descritores de uma imagem de entrada fornecida pelo banco de dados, uma camada oculta, e uma camada de saída consistindo em 4 neurônios (um para cada classe: 1 - direita, 2 - esquerda, 3 - em cima, 4 - embaixo), de acordo com a Figura 4.9. Um subconjunto de exemplos é então utilizado para treinar a rede por meio de um algoritmo apropriado, são eles: Levenberg-Marquadt (LM), *Resilient Backpropagation* (RPROP), *Conjugate Gradient with Powell-Beale Restarts* (CGP), *BFGS Quasi-Newton* (BFG), *Powell-Beale Conjugate Gradient Backpropagation* (CGB) e *Scaled Conjugate Gradient* (SCG).

Generalização: Em seguida, logo após o treinamento, o desempenho de reconhecimento dos padrões da rede treinada é testado com dados não apresentados anteriormente. Então, por exemplo, uma imagem de entrada é apresentada para a rede, sendo que agora não lhe é informado a classe correspondente a esta imagem específica. O desempenho da rede é então estimado comparando-se o reconhecimento da imagem de entrada e a real classe a qual ela pertence.

Para este trabalho, o projeto da rede neural utilizou em sua arquitetura as redes MLP (*MultiLayer Perceptrons*) com algumas variações na topologia (número de neurônios e camadas), na configuração de parâmetros e nos algoritmos de treinamento. Os resultados apresentados no capítulo 5 mostram a adequação destas combinações para os diversos casos apresentados.

4.4 Conclusão

Este capítulo apresentou detalhadamente a construção de um sistema completo para a estimação da direção do olhar, baseando-se em toda a teoria exibida nos capítulos anteriores. A partir do detalhamento do sistema apresentado, foi possível entender o seu real funcionamento, inclusive de forma organizada (com o diagrama de blocos).

O sistema apresentado é composto de três partes fundamentais, que realizam todo o processo para estimar a direção do olhar. Inicialmente, foi realizado o processamento de baixo nível, através da coleta de dados (captura de imagens) de forma criteriosa e do pré-processamento destes dados. Em seguida, no processamento de nível intermediário, foram aplicados os conceitos de processamento digital de imagens relacionados à segmentação e seleção de atributos, e à construção de um banco de dados com informações realmente relevantes das imagens selecionadas, minimizando assim, certas ambiguidades e erros no decorrer do sistema. Finalmente, foi realizado o processamento de alto nível, utilizando-se as redes neurais artificiais

para realizar o treinamento e a classificação dos padrões (direita, esquerda, em cima, embaixo) para a correta estimação da direção do olhar.

O capítulo 5 implementa todo o sistema apresentado, realizando inclusive, testes de desempenho com os algoritmos de detecção da face, detecção do olho e seleção do algoritmo de treinamento para rede neural, assim como os melhores resultados de classificação, utilizando a transformada *Wavelet* como forma de selecionar descritores mais adequados das imagens.

Capítulo 5

Análise de Resultados

Este capítulo reúne os resultados mais significativos deste trabalho. Aplicando-se os conceitos sobre processamento de imagens, transformadas *Wavelets* e redes neurais artificiais, foi realizada uma série de experimentos computacionais com o intuito de avaliar todo o sistema proposto no capítulo 4 .

A primeira parte deste capítulo apresenta os experimentos realizados no processo de segmentação de imagens. Em seguida, na segunda parte, os experimentos tratam do reconhecimento de padrões utilizando um classificador neural, a fim de estimar a direção do olhar. Construiu-se uma base de dados com imagens de 45 usuários, utilizadas para testar e validar todas as partes constituintes do sistema proposto.

5.1 Segmentando imagens

Grande parte do trabalho está vinculada à segmentação de imagens. O problema em questão consiste em segmentar imagens de forma eficiente e adequada ao sistema proposto. Assim, esta seção tem como objetivo avaliar o desempenho dos algoritmos e técnicas de processamento digital de imagens utilizadas para solução deste problema.

5.1.1 Detectando a face

É necessário que o algoritmo para detecção da face utilizado seja bastante eficiente, de acordo com o cenário indicado no capítulo 4 , pois o funcionamento do restante do sistema depende especialmente desta etapa. Assim, neste experimento, foram excedidas todas as distâncias e orientações recomendadas pelo sistema proposto, a fim de garantir com precisão o seu funcionamento nas situações recomendadas. Os testes foram realizados da seguinte forma: Amostras com imagens de 10 usuários formam conjuntos de exemplos, em que nestes foram aplicadas 4 métricas para avaliar o desempenho do algoritmo, são elas:

Distância entre a tela do computador e o usuário: Esta métrica mede o desempenho (percentual de acertos) do algoritmo para detecção da face em contraste com variações de distância. A captura das imagens obedeceu quatro possibilidades

de distâncias entre a face e a tela do computador, são elas: $d1 = 60$ cm, $d2 = 50$ cm, $d3 = 70$ cm e $d4 = 100$ cm. Lembrando que $d1$ é a distância adequada ao sistema proposto, $d2$, $d3$ e $d4$ são excessos que não devem ocorrer na prática, mas servem para medir a confiabilidade do algoritmo. A Figura 5.1 é um exemplo de aplicação da métrica utilizada:

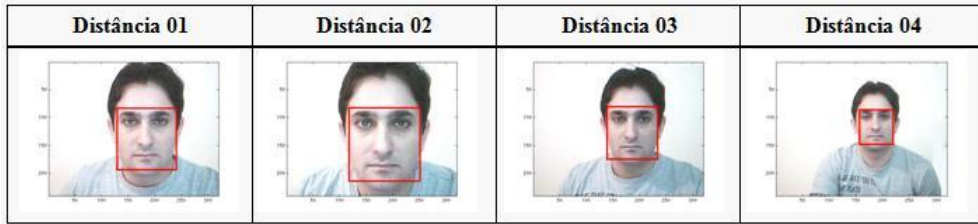


Figura 5.1: Detecção da face utilizando a métrica distância.

A Tabela 5.1 mostra os resultados obtidos para a amostra selecionada: são 10 usuários, sendo 2 imagens por usuário para cada uma das distâncias utilizadas, totalizando a análise de 80 imagens.

Tabela 5.1: Resultados de aplicação da métrica distância.

Métrica	Distância 1	Distância 2	Distância 3	Distância 4
Média de acertos	100%	100%	100%	95%

A distância recomendada é de $d1 = 60$ cm e variações máximas destas em torno de 10%. Dessa forma, verifica-se que ainda assim, excedendo tais distâncias em $d2$, $d3$ e $d4$, o algoritmo foi capaz de detectar a face com acertos sempre iguais ou superiores a 95% dos casos, satisfazendo com tranquilidade este requisito para seu uso no sistema proposto.

Orientação da face em relação à câmera: Esta métrica consiste em medir se a detecção da face é invariante a pequenas variações de orientação espacial. É um fator importante, pois na prática pequenas variações deste tipo devem acontecer com frequência. A Figura 5.2 mostra o primeiro exemplo de aplicação desta métrica de desempenho, onde o usuário é submetido a duas rotações do tipo RIP (*rotation in plane*) de: $+/- 15^\circ$ e $+/- 30^\circ$.

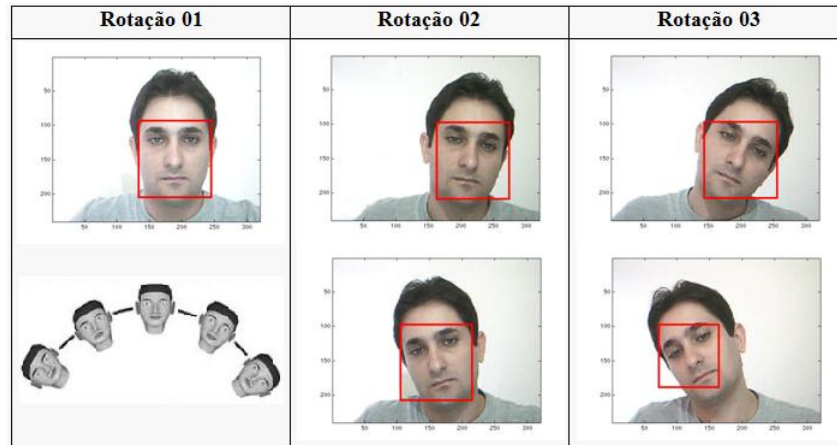


Figura 5.2: Detecção da face utilizando a métrica orientação espacial RIP.

A Figura 5.3 mostra o segundo exemplo de aplicação desta métrica de desempenho, onde o usuário é submetido a duas rotações do tipo ROP (*rotation on plane*) de: $+/- 15^\circ$ e $+/- 30^\circ$.

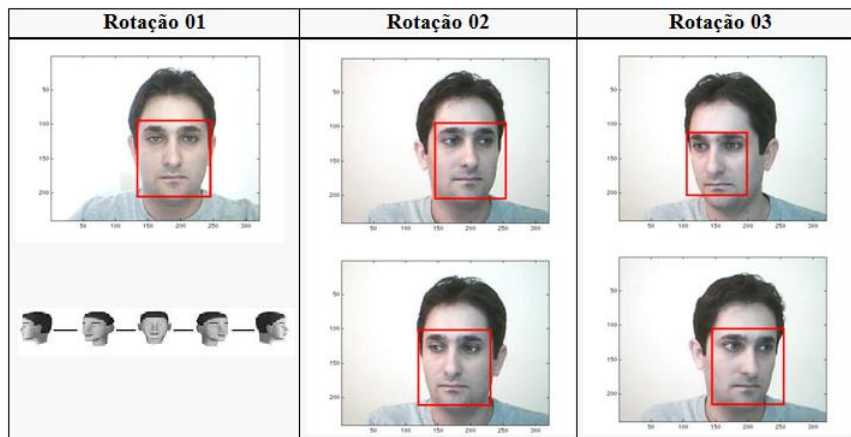


Figura 5.3: Detecção da face utilizando a métrica orientação espacial ROP.

A Tabela 5.2 mostra os resultados obtidos para a amostra selecionada: são 10 usuários, sendo 2 imagens por usuário para cada uma das rotações (RIP e ROP) utilizadas, totalizando 120 imagens analisadas.

Tabela 5.2: Resultados de aplicação da métrica orientação espacial.

Métrica	Rotação 1	Rotação 2	Rotação 3
Média de acertos (RIP)	100%	100%	95%
Média de acertos (ROP)	100%	100%	100%

Variações de luminosidade do ambiente: Controlar fisicamente a atividade luminosa nos mais diversos ambientes não é uma tarefa recomendável. Pode-se contornar estes problemas com *softwares* específicos ou câmeras especiais, estas últimas encarecem bastante o sistema. Desta forma, esta métrica tem como objetivo avaliar o desempenho da detecção da face em condições adversas de luminosidade. A Figura 5.4 mostra um exemplo de aplicação desta métrica de desempenho, onde o usuário é submetido a diferentes níveis de luminosidade (1-ambiente, 2 - Perda de 40%, 3 - Perda de 90%) e abaixo são mostrados seus respectivos histogramas em escala de cinza.

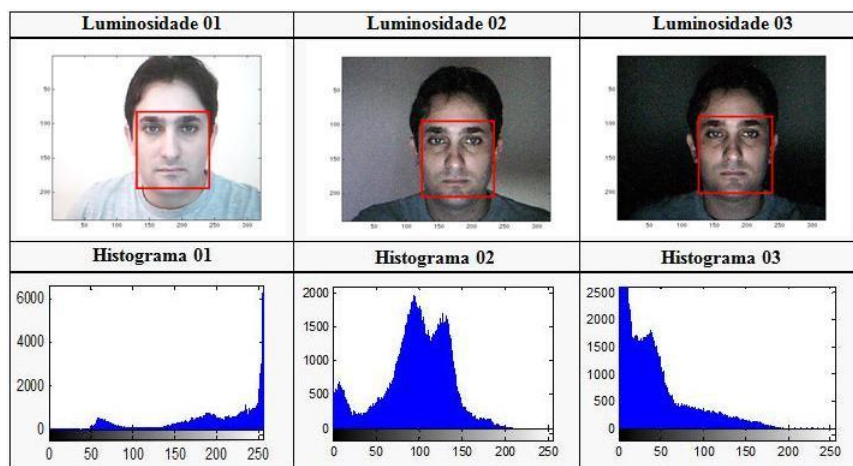


Figura 5.4: Detecção da face utilizando a métrica luminosidade.

A Tabela 5.3 mostra os resultados obtidos para a amostra selecionada: são 10 usuários, sendo 2 imagens por usuário para cada nível de luminosidade do ambiente, totalizando 60 imagens analisadas.

Tabela 5.3: Resultados de aplicação da métrica luminosidade.

Métrica	Luminosidade 1	Luminosidade 2	Luminosidade 3
Média de acertos	100%	100%	93,3%

Observa-se que apenas houve dificuldades de detecção para a luminosidade 3. Mesmo assim, neste nível, os resultados foram de 95%. Isto garante pleno funcionamento do algoritmo em condições reais. Nota-se que grandes variações nos histogramas das imagens são comuns neste tipo de experimento.

Diversidade do ambiente proposto: A diversidade do ambiente de trabalho pode ser um grande problema para a detecção da face, principalmente para algoritmos que fazem uso de geometria e análise da textura das imagens a fim de fazer tal detecção. A Figura 5.5 mostra um exemplo de aplicação desta métrica de desempenho, onde o usuário é submetido a diversos tipos de ambiente.

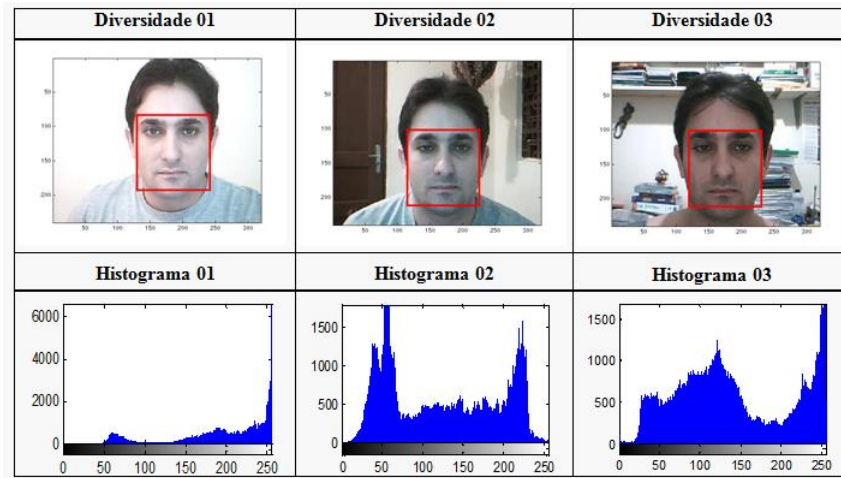


Figura 5.5: Detecção da face utilizando a métrica diversidade.

A Tabela 5.4 mostra os resultados obtidos para a amostra selecionada: são 10 usuários, sendo 2 imagens por usuário para cada nível de diversidade do ambiente, totalizando 60 imagens. Pode-se observar que neste experimento o histograma das imagens em níveis de cinza varia bastante e, mesmo assim, os resultados obtidos estão entre 95% – 100%.

Tabela 5.4: Resultados de aplicação da métrica diversidade.

Métrica	Diversidade 1	Diversidade 2	Diversidade 3
Média de acertos	100%	95%	95%

Por fim, a Figura 5.6 apresenta exemplos de detecção da face utilizado nas condições adequadas, são elas: distância entre a câmera e o usuário em torno de 60 cm, luminosidade ambiente, pouca diversidade no ambiente e rotações (RIP/ROP) mínimas ($\pm 10^\circ$).

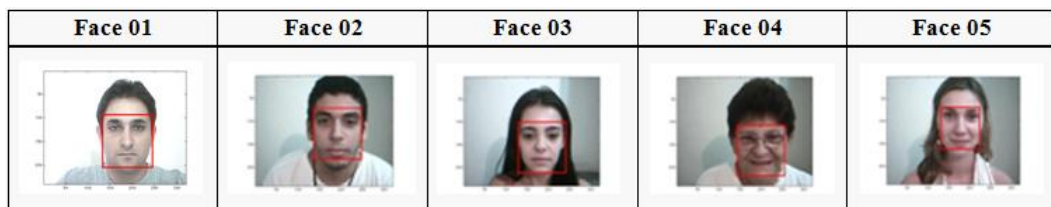


Figura 5.6: Exemplos de detecção da face no ambiente proposto.

Nestas condições, o experimento utilizou uma amostra de 10 usuários, sendo utilizadas 2 imagens para cada detecção, resultando em um acerto de 100% dos testes. Todas as imagens utilizadas nestes experimentos foram adquiridas para este trabalho.

5.1.2 Detectando o olho

Depois de avaliar o desempenho da detecção da face com métricas adequadas aos experimentos realizados na seção anterior, agora é a vez da detecção do olho. Neste caso não foram utilizadas amostras, mas sim todo o banco de dados construído. Este é composto por imagens de 45 usuários, sendo duas imagens de cada posição do olhar, são elas: em cima, embaixo, esquerda e direita, totalizando 360 imagens. A Figura 5.7 mostra exemplos de detecção do olho para 5 usuários.

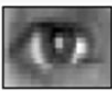
Usuários Posições	Usuário 01	Usuário 02	Usuário 03	Usuário 04	Usuário 05
Em cima					
Embaixo					
Esquerda					
Direita					

Figura 5.7: Exemplos de detecção do olho no ambiente proposto.

A Tabela 5.5 mostra os resultados da detecção do olho. O melhor resultado está na posição em cima, com sua média de acertos igual a 97,8%. Nesta posição, o olho está mais aberto em relação à câmera, facilitando a detecção. Em seguida, estão as posições esquerda e direita, com 97,2% de acertos. Por fim, com 95% de acertos está a posição embaixo, onde ocorrem as maiores dificuldades para detecção, principalmente devido a erros na determinação do seu centro de massa.

Tabela 5.5: Exemplos de detecção do olho.

Posição	em cima	embaixo	esquerda	direita
Média de acertos	97,8%	95%	97,2%	97,2%

5.2 Estimando com redes neurais artificiais

Para a estimação das posições do olhar selecionadas (em cima, embaixo, direita e esquerda), este trabalho fez uso das redes neurais artificiais. Três etapas importantes foram realizadas para obtenção dos resultados, são elas: Criação de um conjunto de treinamento com descritores característicos da informação, escolha de um algoritmo de treinamento para rede neural, e a classificação dos padrões.

5.2.1 Conjunto de treinamento: Extrairando descritores

As imagens utilizadas neste trabalho possuem forma, cor e textura, propriedades estas que representam sobre certa medida, as características das imagens. Descritores de imagens são agrupamentos destas características em um vetor de escalares.

A seção 5.1.2 mostrou como foi realizada a detecção do olho neste trabalho. A partir daí é possível extrair informações relevantes das imagens, a fim de construir um conjunto de treinamento adequado à solução do problema. Uma forma simples de fazer isso, é obter a distribuição de cores das imagens, levando em consideração aspectos geométricos desta distribuição ou histogramas das mesmas. A Figura 5.8 indica a amplitude de cada *pixel* em escala de cinza da imagem de detecção do olho.

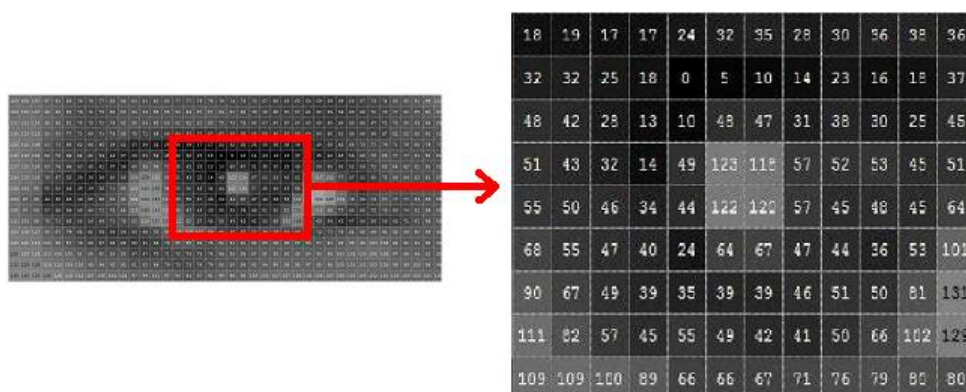


Figura 5.8: Amplitude de cada *pixel* em escala de cinza da imagem de detecção do olho.

A extração de descritores *Wavelets* tem como objetivo selecionar características essencialmente relevantes (uma assinatura dos padrões), somente detectadas com este tipo de ferramenta. As *Wavelets* são funções matemáticas que separam as diferentes componentes do sinal (neste caso as imagens), e extraem cada componente com uma resolução adequada a sua escala. A extração de descritores (coeficientes *Wavelets*) nesta etapa utilizou em sua decomposição, a *Wavelet* mãe db3 (*Daubechies*). Sua escolha se deu devido aos bons resultados em testes empíricos preliminares e outros indicados na literatura.

A Figura 5.9 mostra graficamente um exemplo do procedimento de obtenção dos descritores *Wavelets*. Na primeira coluna, são observadas as imagens da caixa que contém o olho detectado na sua respectiva direção (em cima, embaixo, esquerda e direita). Na segunda coluna, tem-se os histogramas das imagens selecionadas. Na terceira coluna, são apresentados os descritores *Wavelets* no nível 1, seguida da coluna 4 com descritores *Wavelets* no nível 3.

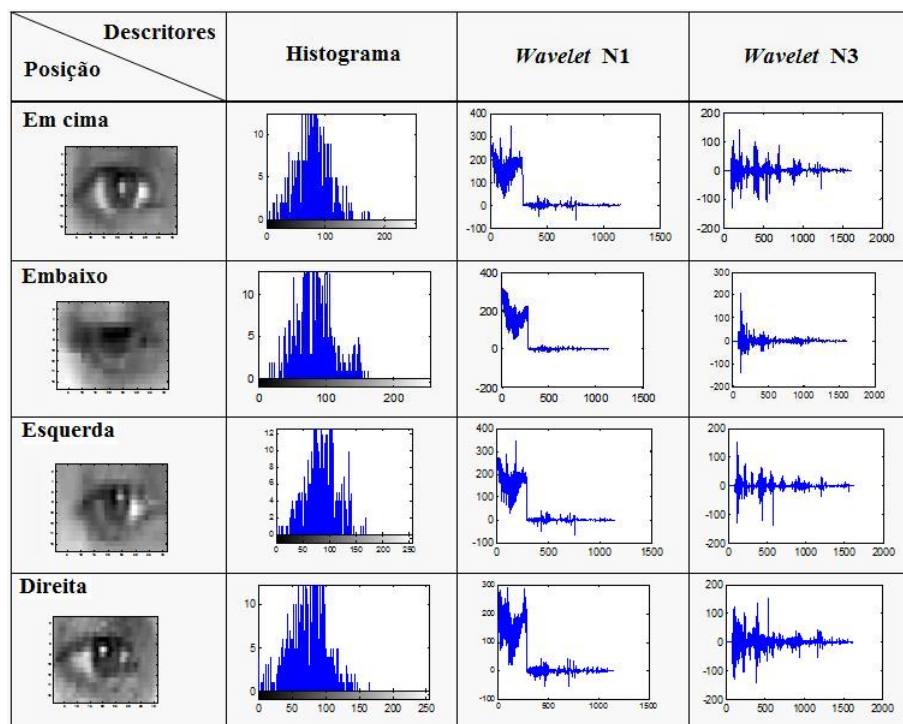


Figura 5.9: Descritores das imagens: histogramas e coeficientes *Wavelets*.

Neste trabalho, foram utilizados os descritores obtidos de nível 3. O resultado deste processo de filtragem utilizando a transformada *Wavelet* ainda passou por mais um processo de seleção. Uma janela de tamanho fixo que vai de 100 a 500 no eixo horizontal seleciona todos os coeficientes ali presentes. Esta janela deve conter informações suficientes para a construção de um conjunto de treinamento qualificado.

Inicialmente, a grande vantagem de aplicação desta transformada é a possibilidade de redução de tamanho do conjunto de treinamento a ser utilizado pela rede neural. A Tabela 5.6 mostra um exemplo real dos ganhos em relação ao tamanho do vetor de descritores.

Tabela 5.6: Vetor de descritores.

Tamanho do vetor	Uma imagem (697 x 1)	todas as imagens (697 x 360)
Sem <i>Wavelets</i>	697	250920
Com <i>Wavelets</i>	400	14400

Com a aplicação da transformada *Wavelet*, observa-se uma redução no vetor de descritores em mais de 17 vezes no tamanho do conjunto de treinamento quando comparado com o mesmo processo apenas utilizando informações de cor *pixel* a *pixel*. Esta redução possibilita uma maior rapidez no treinamento da rede neural. Além disso, a esparsidade gerada pela aplicação das *Wavelets* no sinal, proporciona

um custo computacional para sua manipulação ainda menor.

5.2.2 Estimando a direção do olhar

Foram realizados testes de desempenho com os algoritmos de treinamento dispostos no capítulo 3 com o objetivo de escolher uma boa opção de treinamento para a rede neural, são eles: Levenberg-Marquadt (LM), *Resilient Backpropagation* (RPROP), *Conjugate Gradient with Powell-Beale Restarts* (CGP), *BFGS Quasi-Newton* (BFG), *Powell-Beale Conjugate Gradient Backpropagation* (CGB), *Scaled Conjugate Gradient* (SCG). Todos são considerados algoritmos rápidos, mas testes empíricos direcionados a um problema em específico podem mostrar vantagens e desvantagens entre eles.

A Figura 5.10 mostra os resultados do teste de desempenho para cada um dos algoritmos utilizados, aplicados a uma amostra selecionada do banco de dados construído.

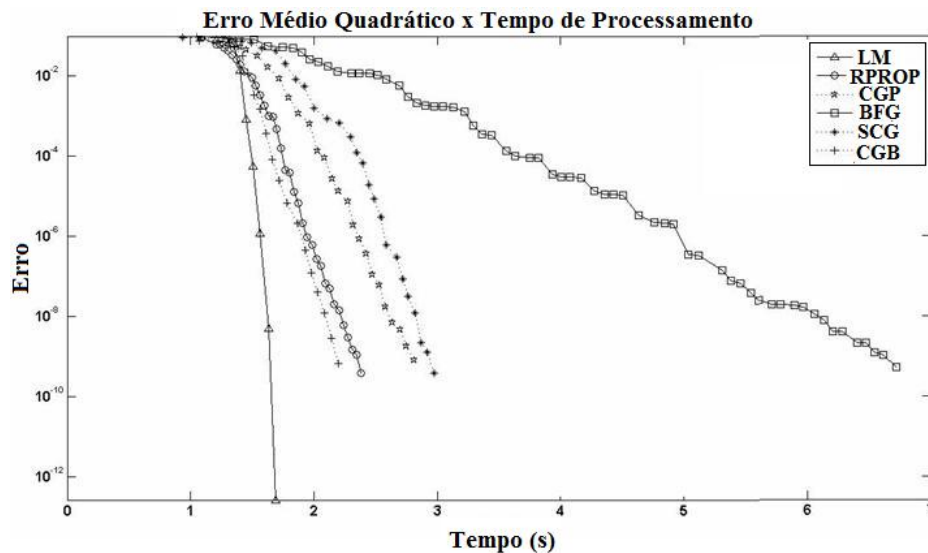


Figura 5.10: Erro médio quadrático em função do tempo de processamento para treinamento da rede em cada algoritmo.

Observa-se que os algoritmos LM, CGB e RPROP obtiveram os melhores resultados. O BFG foi o único que convergiu lentamente, mais de 2 vezes o tempo de qualquer um outro testado. Outro fator importante é que todos os algoritmos convergiram com um erro médio quadrático acima de 10^{-8} , valor este, importante para as aplicações neste trabalho.

Para a implementação do classificador neural, foi utilizado o algoritmo RPROP devido aos bons resultados obtidos no teste de desempenho para treinamento da rede, e principalmente pela sua facilidade de implementação (escrever o algoritmo).

A seguir, são mostrados os resultados mais relevantes, obtidos durante a produção deste trabalho, dispostos em sua ordem de desenvolvimento. Percebe-se que

no decorrer dos resultados existe uma alternância no tamanho do conjunto de treinamento e, conseqüentemente, em sua diversidade, até que se chegue na forma mais adequada. Foram também inseridas as transformadas *Wavelets* como forma de extrair descritores característicos das imagens.

Resultado I

A matriz de confusão disposta na Tabela 5.7 mostra os resultados de classificação obtidos para a seguinte configuração do sistema:

Número de usuários = 10;
 Posições analisadas = 4 (em cima, embaixo, esquerda, direita);
 Quantidade de imagens por posição = 50;
 Conjunto de treinamento = 152 imagens;
 Conjunto de validação = 48 imagens;
 Total de imagens analisadas = 200;
 Arquitetura da rede: MLP, 1-2-1 , RPROP;
 Critérios de parada (Erro ou Número de épocas) = 10^{-5} , 100;

Tabela 5.7: Matriz de confusão para a rede MLP-1.

MLP-1	Em cima	Embaixo	Esquerda	Direita	Acertos
Em cima	10	1	0	1	83,3%
Embaixo	0	10	1	1	83,3%
Esquerda	0	2	10	0	83,3%
Direita	1	0	0	11	91,7%

Estes resultados foram apresentados em [PEIXOTO 2009a]. As posições em cima, embaixo e esquerda obtiveram acertos na ordem de 83,3%, seguidos dos 91,7% da posição direita. Nota-se que os maiores problemas de classificação estão associados a confundir a posição esquerda como sendo embaixo.

Resultado II

A matriz de confusão disposta na Tabela 5.8 mostra os resultados de classificação obtidos para a seguinte configuração do sistema:

Número de usuários = 1;
 Posições analisadas = 4 (em cima, embaixo, esquerda, direita);
 Quantidade de imagens por posição = 50;
 Conjunto de treinamento = 152 imagens;
 Conjunto de validação = 48 imagens;
 Total de imagens analisadas = 200;
 Arquitetura da rede: MLP, 1-2-1 , RPROP;
 Critérios de parada (Erro ou Número de épocas) = 10^{-5} , 100;

Tabela 5.8: Matriz de confusão para a rede MLP-1.

MLP-1	Em cima	Embaixo	Esquerda	Direita	Acertos
Em cima	11	0	0	1	91,7%
Embaixo	0	10	2	0	83,3%
Esquerda	0	1	11	0	91,7%
Direita	1	0	0	11	91,7%

Estes resultados foram apresentados em [PEIXOTO 2009b]. As posições em cima, esquerda e direita obtiveram acertos na ordem de 91,7%, seguidos dos 83,3% da posição embaixo. Nota-se que os maiores problemas de classificação estão associados a confundir a posição embaixo como sendo esquerda.

Resultado III

A matriz de confusão disposta na Tabela 5.9 mostra os resultados de classificação obtidos para a seguinte configuração do sistema:

Número de usuários = 45;
 Posições analisadas = 4 (em cima, embaixo, esquerda, direita);
 Quantidade de imagens por posição = 200;
 Conjunto de treinamento = 440 imagens;
 Conjunto de validação = 360 imagens;
 Total de imagens analisadas = 800;
 Arquitetura da rede: MLP, 1-2-1 , RPROP;
 Critérios de parada (Erro ou Número de épocas) = 10^{-4} , 100;

Tabela 5.9: Matriz de confusão para a rede MLP-2.

MLP-2	Em cima	Embaixo	Esquerda	Direita	Acertos
Em cima	81	2	3	4	90,0%
Embaixo	2	83	4	1	92,2%
Esquerda	2	3	83	2	92,2%
Direita	5	1	2	82	91,1%

Estes resultados mostram a qualidade do sistema proposto com o objetivo de estimar a direção do olhar. As posições embaixo e esquerda obtiveram os melhores resultados com acertos na ordem de 92,2%, seguidos da posição direita com 91,1% e da posição em cima com 91,7%. Nota-se que o maior erro ocorre em classificar a posição direita como em cima; visualmente, pode-se ver a semelhança dos padrões na Figura 5.9, confirmando os motivos do erro.

A Tabela 5.8 revela, de forma geral, a melhoria dos resultados de classificação. Isto é devido a três motivos principais: o aumento no tamanho do conjunto de

treinamento, a sua maior diversidade e a seleção de descritores (*Wavelets*) mais especializados das imagens. O resultados I e II não utilizaram descritores *Wavelets*.

5.3 Conclusão

Este capítulo apresentou os principais resultados obtidos a partir deste trabalho. Inicialmente, foi feita uma análise de desempenho dos algoritmos de detecção da face e olho para as várias métricas estabelecidas. Diante dos bons resultados obtidos, foi possível continuar com o desenvolvimento do sistema proposto.

Na segunda parte deste capítulo foram apresentados os resultados da estimação da direção do olhar utilizando redes neurais artificiais. Foram exibidas as configurações utilizadas nas redes neurais, assim como os resultados obtidos em uma matriz de confusão. Foi possível verificar também a evolução dos resultados, uma vez que estes foram publicados em importantes eventos acadêmicos, comprovando assim a eficácia das abordagens.

Capítulo 6

Conclusão

Neste trabalho foi realizado o desenvolvimento de um sistema não-intrusivo para estimação da direção do olhar. Para isto, foram utilizadas técnicas de processamento digital de imagens, transformadas *Wavelets* e redes neurais artificiais.

Foi realizada uma série de estudos relacionados ao desenvolvimento de sistemas para interação homem-máquina, mais especificamente, aqueles que utilizam os movimentos dos olhos para este objetivo. Verificou-se um amplo desenvolvimento dos métodos intrusivos e uma lacuna no desenvolvimento dos métodos não-intrusivos, esta última devida, principalmente, à dificuldade de se trabalhar com as inúmeras variáveis, características deste tipo de problema.

Foram consideradas três etapas essenciais no desenvolvimento deste trabalho: o processamento de baixo nível, responsável pela aquisição e o pré-processamento das imagens; o processamento de nível intermediário, onde as imagens são analisadas detalhadamente; e o processamento de alto nível, utilizado para o reconhecimento e a interpretação dos padrões de direção do olhar.

Para o adequado funcionamento do sistema proposto, uma série de tarefas foram seguidas na ordem indicada pelo diagrama de blocos apresentado no capítulo 4. Inicialmente, de forma não-intrusiva, foi realizada a aquisição das imagens utilizando uma simples *webcam* acoplada a um computador pessoal (*notebook*) que possuía instalado o sistema operacional Ubuntu Linux, além do *software* de desenvolvimento Matlab (versão estudante). Posteriormente, iniciou-se a segunda etapa de desenvolvimento: foram aplicados vários algoritmos nas imagens selecionadas, com o objetivo de segmentá-las e extrair descritores relevantes para a construção de uma base de dados. Dentre os algoritmos mais importantes ligados à segmentação, foram considerados o algoritmo para detecção da face, que utiliza a transformada SQMT (*Successive Mean Quantization Transform*), para selecionar a face do usuário, retirando toda a informação adicional inserida pelo ambiente, e o algoritmo para detecção do olho que foi uma contribuição deste trabalho, desenvolvido com base em técnicas de geometria facial, limiarização, detecção de objetos e centro de massa das imagens. Para a construção de uma base de dados eficiente com informações dos olhos, aplicou-se na extração de descritores das imagens o uso das transformadas *Wavelets*, realizando sua decomposição com a *Wavelet* mãe db3 (Daubechies) em até três níveis. Na terceira e última etapa, foi implementado um classificador neural *Multilayer Perceptrons* utilizando o algoritmo RPROP (*Resilient Backpropagation*)

em diversas configurações, com o objetivo de estimar a direção do olhar.

Os resultados deste trabalho foram apresentados em duas categorias. Inicialmente, foi realizada a segmentação de imagens, com a validação do algoritmo de detecção da face, a partir de métricas de desempenho aplicadas às amostras selecionadas. Em seguida, foi realizada a validação da detecção do olho, comprovando a eficácia do sistema proposto neste trabalho. Na segunda parte, foram analisados os resultados de classificação obtidos com as redes neurais. Verificou-se uma melhoria considerável nos resultados de classificação, uma vez que houve um crescimento no tamanho do conjunto de treinamento, o aumento de sua diversidade e a utilização das transformadas *Wavelets* para extração de descritores característicos das imagens. A aplicação das *Wavelets* possibilitou construir um conjunto de treinamento mais otimizado e adequado ao problema.

Desenvolver sistemas não-intrusivos para estimação da direção do olhar é um enorme desafio da atualidade. Neste trabalho foi possível estabelecer uma coordenação de tarefas a fim de cumprir este objetivo. O resultado foi um sistema que envolve *hardware* e *software* de forma funcional e qualificada, concatenando uma série de recursos computacionais, que envolvidos podem estimar a direção do olhar de forma satisfatória. Isto possibilita uma gama de aplicações para interação homem-máquina. Além disso, pode-se dizer que foi elaborada uma abordagem autêntica e bem sucedida para a solução do problema proposto.

Como propostas à continuidade do trabalho, novas investigações devem ser realizadas, objetivando principalmente a aplicação de outras técnicas de processamento de imagens e de redes neurais artificiais. Observa-se que a organização do sistema proposto permite a facilidade necessária para se substituir partes (blocos) do sistema de forma independente, desde que sejam mantidas algumas premissas de compatibilidade.

Referências Bibliográficas

- AHUJA, M. H. YANG D. J. KRIEGMAN N. (2002), ‘Detecting faces in images: A survey’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(1).
- ARA, N. (1998), ‘Face detection and recognition using hidden markov models’, *International Conference on Image Processing* pp. 141–145.
- BEALE, M. T. HAGAN H. B. DEMUTH M. H. (1996), *Neural network design*, Bookman, Boston, MA.
- BHATIA, P. BHARTI S. GUPTA R. (2009), ‘Comparative analysis of image compression techniques: A case study on medical images’, *International Conference on Advances in Recent Technologies in Communication and Computing* pp. 820–822.
- BRAUN, M. RIEDMILLER H. (1993), A direct adaptative method for faster back-propagation learning: the rprop algorithm, *em* ‘Proceedings’, IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS, San Francisco, USA, pp. 586 - 591.
- CARPENTER, M. A. JUST P. A. (1980), ‘A theory of reading: from eye fixation to comprehension’, *Psychol Rev* (87), 329–354.
- CASTLEMAN, K. R. (1996), *Digital Imaging Processing*, 1ª edição, Ed. Prentice Hall.
- CLAESSON, M. NILSSON J. O. NORDBERG I. (2007), ‘Face detection using local smqt features and split up snowclassifier’, *ICASSP* **2**, 589–592.
- CLAESSON, M. NILSSON M. DAHL I. (2005), ‘The successive mean quantization transform’, *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* **4**, 429–432.
- DAUNICHT, D. OTT W. J. (1992), ‘Eye movement measurement with the scanning laser ophthalmoscope’, *Clin. Vision Sci.* (7), 551–556.
- DEL, L. HUCHUAN Y. Z. WEI Y. (2007), ‘Eye detection based on rectangle features and pixel-pattern-based texture features’, *Department of Electronic Engineering, Dalian University of Technology*.

- FLICKNER, C. H. MORIMOTO D. KOONS A. AMIT M. (2000), ‘Pupil detection and tracking using multiple light sources’, *Image and Vision Computing, The Netherlands* **18**(4), 331–336.
- FONSECA, L. M. G. (2000), ‘Processamento digital de imagens’, *INPE - Instituto de Pesquisas Espaciais* p. 111.
- GOMES, J. E. R. QUEIROZ H. M. (2001), ‘Introdução ao processamento digital de imagens’, *Universidade Federal de Campina Grande*.
- GONZALEZ, R. E. WOODS R. C. (2000), *Processamento de Imagens Digitais*, 1ª edição, Edgard Blucher LTDA, São Paulo, SP.
- HAGAN, H. DEMUTH M. BEALE M. (2009), *Neural Network Toolbox*, Copyright by The MathWorks, Inc.
- HAMID, N. A. JALAL K. SARA P. R. (2008), ‘Eye detection algorithm on facial color images’, *Second Asia International Conference on Modelling and Simulation*.
- HAYKIN, S. (2001), *Redes Neurais - princípios e prática*, 2ª edição, Bookman, Porto alegre, RS.
- HOFFMAN, J. E. (1998), ‘Visual attention and eye movements’, *UK: Psychology Press* In H. Pashler, 119–154.
- HONG, H. W. LEE S. KIL Y. HAN S.H. (2001), ‘Automatic face and facial fetures detection’, *Proceedings of Conference on ISIE* pp. 254–259.
- JI, D. WITZNER Q. (2010), In the eye of the beholder:a survey of models for eyes and gaze, *em ‘IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE’*, IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS, pp. 478 - 500.
- KEESE, A. (2009), ‘Teoria de cores’, Desktop Publishing.
- LENT, R. (2002), *Cem Bilhões de Neurônios - Conceitos Fundamentais de Neurociência*, Editora Atheneu.
- LINDA, M. R. HARALICK S. G. (1992), ‘Computer and robot vision’, *Addison-Wesley* **1**, 28–48.
- LOPES, E. C. (2005), ‘Detecção de faces e características faciais’, *Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS* (Relatório Técnico, 45).
- LUGER, G. F. (2004), *Inteligência Artificial, Estrutura e Estratégias para Solução de Problemas Complexos*, 4ª edição, Bookman.
- MALLAT, S. (2009), *A Wavelet Tour of Signal Processing*, 3ª edição, Elsevier Inc.

- MASNADI-SHIRAZI, R. MOHSENI A. SHEIKHI M. A. (2009), 'Efficient compression of wavelet packet ofdm radar signals'.
- MCLAREN, J. M. MENDEL R. W. (1970), 'Adaptive, learning and pattern recognition systems: Theory and applications', *Academic Press, New York* pp. 287–318.
- MENHAJ, M. T. HAGAN M. (1999), 'Training feed-foward networks with the marquardt algorithm', *IEEE Transactions on Neural Networks* **5**(6), 989–993.
- MOLLER, M. F. (1993), 'A scaled conjugate gradient algorithm for fast supervised learning', *Neural Networks Magazine* **6**, 525–533.
- MORETTO, E. G. (2004), 'Rastreamento da posição dos olhos para detecção da direção do olhar', *Universidade Católica Dom Bosco*.
- MULLIGAN, J. B. (1997), 'Recovery of motion parameters from distortions in scanned images. proceedings of the nasa image registration workshop (irw97)', *NASA Goddard Space Flight Center*.
- NETO, H. PISTORI J. J. (2003), 'Utilização de tecnologia adaptativa na detecção da direção do olhar', *Spc Magazine* **2**(2).
- OGORMAN, L. (1995), 'Document image analysis', *Los Alamitos: IEEE Computer Society Press*.
- OLIVEIRA, H. M. (2007), *Análise de fourier e wavelets: sinais estacionários e não estacionários*, 1ª edição, Ed. Universitária UFPE.
- OTSU, N. (1979), 'A threshold selection method from gray-level histograms', *IEEE Transactions on Systems, Man and Cybernetics* **SMC** **9**(1), 62–66.
- OTTENSAMER, J. BOBIN J. STARCK R. (2008), 'Compressed sensing in astronomy', *IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING* **2**(5), 718–726.
- PAROLAI, J. J. GALIANA-MERINO J. L. ROSA-HERRANZ S. (2008), 'Seismic *P* phase picking using a kurtosis-based criterion in the stationary wavelet domain', *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING* **46**(11), 3815–3826.
- PEIXOTO, A. M. G. GUERREIRO A. D. D. NETO H. M. (2009a), 'Image processing for eye detection and classification of the gaze direction', *International Joint International Conference on Neural Neural Networks Network Society*.
- PEIXOTO, A. M. G. GUERREIRO A. D. D. NETO H. M. (2009b), 'Non-intrusive system to detect the gaze direction using artificial neural networks', *Anais do IX Congresso Brasileiro de Redes Neurais / Inteligência Computacional - CBRN*.

- PEIXOTO, J. B. A. REGO A. M. G. GUERREIRO A. D. D. NETO H. M. (2009), ‘Otimizando redes neurais artificiais com transformadas wavelets’, *Anais do IX Congresso Brasileiro de Redes Neurais / Inteligência Computacional - CBRN*.
- PINTO, L. F. Q. SILVEIRA L. G. Q. S. JUNIOR F. M. ASSIS E. L. (2009), ‘Analysis and optimization of wavelet-coded communication systems’, **8**(2), 563–657.
- POSNER, M. I. (1980), ‘Orienting of attention’, *Quarterly Journal of Experimental Psychology* (32), 3–25.
- PROTÁZIO, J. M. B. (2002), ‘Análise wavelet aplicada a sinais geofísicos’, *Dissertação de Mestrado - Universidade Estadual de Campinas*.
- RAJALA, A. ALATTAR S. (1999), ‘Facial features localization in front view head and shoulders images’, *IEEE International Conference on Acoustics, Speech and Signal Processing* pp. 3557–3560.
- SCHNABEL, J. E. DENNIS R. B (1983), ‘Numerical methods for unconstrained optimization and nonlinear equations’, *Englewood Cliffs, NJ: Prentice-Hall*.
- SCHNEIDER, H. DEUBEL W. X. (1996), ‘Saccade target selection and object recognition: Evidence for a common attentional mechanism’, *Vision Research* (36), 1827–1837.
- TAVARES, J. M. R. S. (2000), ‘Análise de movimento de corpos deformáveis usando visão computacional’, *Universidade do Porto*.
- VIMAL, P. S. KUMAR R. ARUMUGABATHAN C. (2009), ‘Wavelet based ocular artifact removal from eeg signals using arma method and adaptive filtering’, pp. 667–671.
- WANG, X. Q. G. LIU D. G. Z. LI Z. WANG H. (2009), ‘Object categorization using hierarchical wavelet packet texture descriptors’, *11th IEEE International Symposium on Multimedia* pp. 44–51.
- WARD, R. D. WRIGHT L. M. (1980), ‘Orienting of attention’, *New York. Oxford University Press*.
- WEEKS, M. (2007), *Digital Signal Processing Using MATLAB and Wavelets*, Georgia State University, Infinity Science Press LLC Hingham, Massachusetts.
- YEPENG, G. (2007), ‘Robust eye detection from facial image based on multicue facial information’, *IEEE International Conference on Control and Automation Guangzhou, CHINA*.