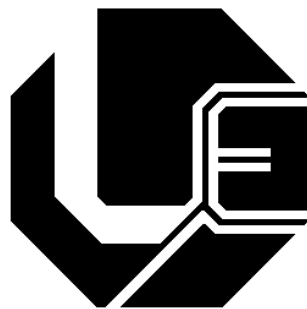


UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



Reconhecimento de Íris utilizando
Algoritmos Genéticos e
Amostragem não Uniforme

Milena Bueno Pereira Carneiro

Dezembro

2010

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**Reconhecimento de Íris utilizando
Algoritmos Genéticos e
Amostragem não Uniforme**

Milena Bueno Pereira Carneiro

Texto da tese de doutorado apresentado à Universidade Federal de Uberlândia, perante a banca de examinadores abaixo, como parte dos requisitos necessários à obtenção do título de Doutor em Ciências.

Banca examinadora:

Antônio Cláudio P. Veiga, Dr - Orientador (UFU)

Adilson Gonzaga, Dr (USP)

Aparecido Nilceu Marana, Dr (UNESP)

Edna Lúcia Flôres, Dra (UFU)

Gilberto Arantes Carrijo, Dr (UFU)

Reconhecimento de Íris utilizando Algoritmos Genéticos e Amostragem não Uniforme

Milena Bueno Pereira Carneiro

Texto da tese apresentada à Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Doutor em Ciências.

Prof. Dr. Antônio Cláudio P. Veiga
Orientador

Prof. Dr. Alexandre Cardoso
Coordenador do curso de Pós-Graduação

*Dedico este trabalho ao meu amado esposo Walterson
e aos meus amados pais Vanderlei e Lúcia.*

Agradecimentos

À Deus, por todas as graças concedidas e por guiar meus caminhos ajudando-me a vencer minhas inseguranças e dificuldades.

Ao meu esposo Walterson pelo amor, carinho e por poder contar sempre com seu intenso apoio e incentivo.

Aos meus pais Vanderlei e Lúcia por todo amor e dedicação com que me educaram e tornaram possíveis todas as minhas conquistas.

Ao meu orientador e amigo Prof. Antônio Cláudio Paschoarelli Veiga pela confiança, apoio e companheirismo de sempre.

Ao amigo Fernando Cordeiro de Castro pelas importantes colaborações com este trabalho.

Aos professores e funcionários da Faculdade de Engenharia Elétrica que direta ou indiretamente contribuíram com minha formação acadêmica e profissional.

Resumo

Carneiro, M. B. P. *Reconhecimento de Íris utilizando Algoritmos Genéticos e Amostragem não Uniforme*, Uberlândia, FEELT-UFU, 2010

O reconhecimento automático de pessoas utilizando-se características da íris é uma eficiente técnica biométrica que está sendo largamente estudada e aplicada em todo o mundo. Diversas etapas de processamento são necessárias para tornar possível a representação e a interpretação da informação contida na íris. Neste trabalho é apresentado o estado da arte de sistemas de reconhecimento de íris onde são citados os trabalhos de maior destaque e as diferentes técnicas empregadas em cada etapa de processamento. São apresentadas implementações de cada etapa de processamento utilizando técnicas tradicionais e, posteriormente, são propostos dois métodos inovadores que têm o objetivo comum de trazer benefícios ao sistema. A primeira etapa de processamento é a localização da região da íris na imagem. O primeiro método proposto neste trabalho apresenta um algoritmo para realizar a localização da íris utilizando os chamados Algoritmos Meméticos. O novo método é comparado a um método clássico e os resultados obtidos demonstram vantagens no que diz respeito à eficiência e ao tempo de processamento. Em uma outra etapa de processamento deve haver uma amostragem de pixels na região da íris, de onde são retiradas as informações utilizadas para diferenciar os indivíduos. Tradicionalmente, esta amostragem é realizada de maneira uniforme ao longo de toda a região da íris. É proposto um método de pré-processamento que sugere uma amostragem não uniforme de pixels na região da íris com o objetivo de selecionar o conjunto de pixels que carregam mais informações da estrutura da íris. A busca por esse conjunto de pixels é realizada utilizando-se Algoritmos Genéticos. A aplicação deste novo método aumenta a eficiência do sistema e ainda possibilita a geração de *templates* binários menores. Neste trabalho é realizado, ainda, um estudo dos chamados Active Shape Models e a sua aplicação para segmentar a região da íris é avaliada. Para a simulação e avaliação dos métodos, foram utilizados dois bancos de imagens de íris públicos e gratuitos: o banco de imagens UBIRIS e o banco de imagens MMU.

Palavras-chave

Reconhecimento de íris, localização de íris, algoritmos genéticos, algoritmos meméticos, amostragem de pixels, Active Shape Models.

Abstract

Carneiro, M. B. P. *Iris Recognition using Genetic Algorithms and Non-Uniform Sampling*, Uberlândia, FEELT-UFU, 2010

The automatic recognition of individuals through the iris characteristics is an efficient biometric technique that is widely studied and applied around the world. Many image processing stages are necessary to make possible the representation and the interpretation of the iris information. This work presents the state of the art in iris recognition systems where the most remarkable works and the different techniques applied to perform each processing stage are quoted. The implementations of each processing stage using traditional techniques are presented and, afterwards, two innovator methods are proposed with the common objective of bringing benefit to the system. The first processing stage should be the localization of the iris region in an eye image. The first method proposed in this work presents an algorithm to achieve the iris localization through the utilization of the called Memetic Algorithms. The new method is compared to a classical method and the obtained results show advantages concerning efficiency and processing time. In another processing stage there must be a pixels sampling from the iris region, from where the information used to differentiate the individuals is extracted. Traditionally, this sampling process is accomplished in an uniform way along the whole iris region. It is proposed a pre-processing method which suggests a non uniform pixels sampling from the iris region with the objective of selecting the group of pixels which carry more information about the iris structure. The search for this group of pixels is done through Genetic Algorithms. The application of the new method improves the efficiency of the system and also, allows the generation of smaller templates. In this work, a study on the called Active Shape Models is also accomplished and its application to perform the iris region segmentation is evaluated. To execute the simulations and the evaluation of the methods, it was used two public and free iris images database: UBIRIS database and MMU database.

Keywords

Iris recognition, iris localization, genetic algorithms, memetic algorithms, pixels sampling, Active Shape Models.

Sumário

Sumário	x
Lista de Figuras	xiii
Lista de Tabelas	xvi
Lista de Códigos	xvii
Simbologia	xix
1 INTRODUÇÃO	1
1.1 Objetivos	2
1.2 Estrutura da Tese	4
2 SISTEMA DE RECONHECIMENTO DE ÍRIS	6
2.1 Etapas de Processamento	8
2.1.1 Etapa de localização	8
2.1.2 Etapa de normalização	10
2.1.3 Etapa de extração e codificação de características . . .	11
2.1.4 Etapa de comparação	13
2.2 Ambiente de Decisão e as Taxas de Erro	14
2.3 Estado da Arte	17
2.4 Considerações Finais do Capítulo	22
3 SISTEMA DE RECONHECIMENTO DE ÍRIS UTILIZADO PARA SIMULAÇÃO	23
3.1 Métodos de Localização	24

3.1.1	Transformada de Hough circular	24
3.1.2	Operador integro-diferencial	27
3.1.3	Detecção das pálpebras e dos cílios	28
3.2	Método de Normalização	30
3.3	Método de Extração e Codificação de Características	35
3.4	Método de Comparação	39
3.5	Bancos de Imagens de Íris	43
3.5.1	Banco de imagens UBIRIS	43
3.5.2	Banco de imagens MMU	44
3.6	Resultados Experimentais	46
3.6.1	Ajuste dos parâmetros para o banco de imagens UBIRIS	47
3.6.2	Ajuste dos parâmetros para o banco de imagens MMU	50
3.7	Conclusões do Capítulo	53
4	MÉTODO PROPOSTO PARA LOCALIZAÇÃO DA ÍRIS	55
4.1	Implementação do Método Proposto usando Algoritmos Meméticos	58
4.2	Mecanismo de Busca Local Implementado	60
4.3	Definição dos Parâmetros	64
4.4	Resultados Experimentais	66
4.4.1	Avaliando o método proposto	66
4.4.2	Comparação entre o método proposto e o método de Wildes	67
4.5	Conclusões do Capítulo	68
5	MÉTODO PROPOSTO PARA AMOSTRAGEM NÃO UNIFORME DE PIXELS NA REGIÃO DA ÍRIS	70
5.1	Método Proposto	73
5.2	Implementação do Método Proposto usando Algoritmos Genéticos	75
5.2.1	Definição da função de aptidão	81
5.3	Resultados Experimentais	82
5.3.1	Resultados com UBIRIS	82
5.3.2	Resultados com MMU	87
5.4	Conclusões do Capítulo	91
6	UTILIZAÇÃO DE ACTIVE SHAPE MODELS PARA SEGMENTAÇÃO DA ÍRIS	93
6.1	Etapa de Treinamento	95

6.1.1	Marcação das formas nas imagens de treinamento	97
6.1.2	Obtenção das imagens em multi-resolução	100
6.1.3	Obtenção do perfil de um ponto da forma	101
6.1.4	Obtenção da estatística do perfil	103
6.1.5	Obtenção da matriz de pesos	105
6.1.6	Alinhamento das formas do conjunto de treinamento .	106
6.1.7	Obtenção da estatística das formas	108
6.2	Etapas de Busca da Forma em uma Imagem	115
6.2.1	Estimativa da forma inicial	116
6.2.2	Obtenção do deslocamento desejado	117
6.2.3	Obtenção dos parâmetros para modificar a pose da forma	120
6.2.4	Obtenção dos parâmetros para deformar o contorno da forma	121
6.2.5	Aplicação dos parâmetros encontrados para a modifi- cação da pose e para a deformação da forma	122
6.2.6	Efeito da utilização de imagens em multi-resolução . .	123
6.3	Pré-Processamento das Imagens de Íris	126
6.4	Resultados da Utilização de ASM para Segmentar a Região da Íris	134
6.5	Avaliação dos Resultados Obtidos	139
6.6	Conclusões do Capítulo	140
7	CONCLUSÕES, CONTRIBUIÇÕES E SUGESTÕES DE TRA-	
	BALHOS FUTUROS	148
7.1	Resumo das Implementações e dos Resultados	148
7.2	Conclusões e Contribuições	151
7.3	Sugestões de Trabalhos Futuros	153
7.4	Publicações	155
	Referências Bibliográficas	156
	Apêndice	166
A	Introdução a Algoritmos Genéticos e Meméticos	166
A.1	Algoritmos Genéticos	166
A.2	Algoritmos Meméticos	171
A.3	Definição de Parâmetros para AG e AM	173
B	Método de Otsu para Limiarização	176

Lista de Figuras

2.1	<i>Diagrama esquemático do sistema de verificação biométrica.</i>	7
2.2	<i>Etapas de processamento do sistema de reconhecimento de íris.</i>	8
2.3	<i>Imagem de um olho do banco de imagens UBIRIS [63].</i>	9
2.4	<i>Ambiente de decisão: distribuições de distância de Hamming geradas a partir de comparações intra-classe e inter-classes.</i>	15
3.1	<i>Acumulador de votos.</i>	25
3.2	<i>a)Imagem de um olho; b)Mapa de bordas correspondente; c)Mapa de bordas utilizando somente o gradiente vertical; d)Mapa de bordas utilizando somente o gradiente horizontal.</i>	27
3.3	<i>Estágios do processo de exclusão das pálpebras. a)Imagem original proveniente do banco de imagens CASIA [1]; b)Retas detectadas utilizando-se a Transformada de Hough Linear correspondente às bordas das pálpebras superior e inferior; c)Retas horizontais que interceptam as primeiras no ponto de borda da íris mais próximo à pupila; d)Exclusão das regiões acima da reta horizontal referente à pálpebra superior e abaixo da referente à pálpebra inferior.</i>	30
3.4	<i>Representação retangular da região da íris proposta por John Daugman [15, 16, 17, 18].</i>	31
3.5	<i>Esboço do processo de normalização com resolução radial igual a 10 pixels e resolução angular igual a 40 pixels.</i>	33
3.6	<i>a)Pixels selecionados na imagem original para a geração da representação retangular da íris; b)Representação retangular da região da íris com dimensões 18 x 240 pixels; c)Máscara de ruídos: a parte branca representa as regiões que sofrem interferência das pálpebras e cílios (dimensões: 18 x 240 pixels).</i>	34
3.7	<i>Par de filtros de Gabor em quadratura.</i>	36
3.8	<i>Função de transferência da componente de simetria par do filtro de Gabor com uma largura de faixa grande. As duas gaussianas que formam a função se sobrepõem na origem resultando em uma componente dc significativa.</i>	37

3.9	<i>Plano complexo utilizado para quantização.</i>	39
3.10	a) <i>Representação normalizada da íris com dimensões 18 x 240 pixels; b) Máscara de ruídos com dimensões 18 x 240 pixels; c) Template binário com dimensões 18 x 480 pixels, gerado à partir da codificação de cada pixel da representação normalizada da íris, sendo que cada pixels é codificado por dois bits; d) Codificação da máscara de ruídos com dimensões 18 x 480 (cada pixel da máscara de ruídos é codificado por dois bits).</i>	40
3.11	<i>Representação de 1 deslocamento onde são deslocados sempre dois bits de cada vez, pois cada pixel da representação normalizada da íris é codificado por dois bits. Deve-se considerar somente a menor distância de Hamming que, neste caso, é igual a 0,28 e representa o melhor casamento entre os dois templates comparados.</i>	42
3.12	<i>Exemplos de imagens de íris do banco de imagens UBIRIS.</i>	44
3.13	<i>Exemplos de imagens de íris do banco de imagens MMU.</i>	45
4.1	<i>Resultado da aplicação do algoritmo de detecção de bordas de Canny.</i>	56
4.2	<i>Estrutura do cromossomo utilizado pelo algoritmo memético.</i>	59
4.3	<i>Problema de mínimo local que frequentemente ocorre na execução do AG.</i>	61
4.4	<i>Divisão do círculo em quadrantes e direção das linhas diagonais a serem selecionadas durante a busca local.</i>	62
4.5	<i>Ilustração do procedimento de busca local.</i>	65
5.1	<i>Estrutura do cromossomo utilizado pelo algoritmo genético.</i>	76
5.2	<i>Exemplo de uma solução do AG.</i>	78
5.3	<i>Mutação trocada (Swap mutation).</i>	79
5.4	<i>Ilustração da troca de até três pares de genes.</i>	80
5.5	<i>Soluções do algoritmo genético para o banco de imagens UBIRIS.</i>	85
5.6	<i>Curvas ROC para o banco de imagens UBIRIS.</i>	87
5.7	<i>Soluções do algoritmo genético para o banco de imagens MMU.</i>	89
5.8	<i>Curvas ROC para o banco de imagens MMU.</i>	91
6.1	<i>Marcação dos pontos da forma.</i>	98
6.2	<i>Retas de referência.</i>	99
6.3	<i>Ilustração da pirâmide de imagens obtida por sucessivas suavizações e sub-amostragens de uma imagem [12].</i>	100
6.4	<i>Máscara gaussiana [12].</i>	101
6.5	<i>Uma imagem em diversos níveis de resolução da pirâmide de imagens.</i>	102
6.6	<i>Procedimento para obtenção do perfil dos pontos da forma.</i>	103
6.7	<i>Pontos do perfil de cada ponto da forma.</i>	104

6.8	<i>Procedimento para alinhar um conjunto de formas.</i>	107
6.9	<i>Conjunto de 30 formas desalinhadas.</i>	108
6.10	<i>Conjunto de 30 formas alinhadas.</i>	109
6.11	<i>Procura ao longo do perfil de busca pelo sub-perfil que fornece o melhor casamento com o perfil da forma [10].</i>	119
6.12	<i>Efeito de se utilizar a mesma quantidade de pixels para gerar o perfil de cada ponto da forma [12].</i>	125
6.13	<i>Lado esquerdo: Imagem original; Meio: Imagem cortada; Lado direito: Resultado do primeiro passo de pré-processamento.</i>	128
6.14	<i>Resultado do segundo passo de pré-processamento.</i>	130
6.15	<i>Função de transformação de intensidade implementada para o alargamento de contraste.</i>	131
6.16	<i>Lado esquerdo: Imagem original; Lado direito: Imagem binarizada pelo método de Otsu.</i>	133
6.17	<i>Lado esquerdo: Imagem original; Lado direito: Resultado final do algoritmo de pré-processamento.</i>	135
6.18	<i>Exemplos de resultados de segmentação correta. Lado esquerdo: forma inicial; Lado direito: forma final obtida pelo ASM.</i>	142
6.19	<i>Exemplos de resultados de erro na segmentação da pálpebra superior.</i>	143
6.20	<i>Exemplos de resultados de erro na segmentação da pupila.</i>	144
6.21	<i>Exemplos de resultados de erro na segmentação da pálpebra superior e da pupila.</i>	145
6.22	<i>Exemplos de resultados de erro na segmentação da borda externa da íris.</i>	146
6.23	<i>Exemplos de resultados de erros graves.</i>	147
A.1	<i>Representação do cruzamento de um ponto de corte</i>	170
A.2	<i>Representação da mutação simples.</i>	170
A.3	<i>Fluxograma de um algoritmo genético genérico.</i>	171
A.4	<i>Fluxograma de um algoritmo memético genérico.</i>	174

Lista de Tabelas

3.1	<i>Ajuste da resolução da representação retangular da íris para o banco de imagens UBIRIS. Os valores da segunda linha representam a resolução radial em número de pixels e os valores da segunda coluna representam a resolução angular em número de pixels. Os demais valores da tabela são as taxas de erro obtidas.</i>	49
3.2	<i>Ajuste dos parâmetros do filtro Log-Gabor para o banco de imagens UBIRIS. Os valores da segunda linha representam a largura de banda do filtro (σ_{Onf}) e os valores da segunda coluna representam o comprimento de onda central (minWaveLength). Os demais valores da tabela são as taxas de erro obtidas.</i>	50
3.3	<i>Ajuste da resolução da representação retangular da íris para o banco de imagens MMU. Os valores da primeira linha representam a resolução radial em pixels e os valores da primeira coluna representam a resolução angular em pixels. Os demais valores da tabela são as taxas de erro obtidas.</i>	52
3.4	<i>Ajuste dos parâmetros do filtro Log-Gabor para o banco de imagens MMU. Os valores da segunda linha representam a largura de banda do filtro (σ_{Onf}) e os valores da segunda coluna representam o comprimento de onda central (minWaveLength). Os demais valores da tabela são as taxas de erro obtidas.</i>	53
4.1	<i>Comparação dos resultados obtidos com o método proposto e o de Wildes.</i>	68
5.1	<i>Resumo dos resultados das simulações utilizando o banco de imagens UBIRIS.</i>	84
5.2	<i>Tabulações das curvas ROC resultantes dos experimentos utilizando o banco de imagens UBIRIS.</i>	87
5.3	<i>Resumo dos resultados das simulações utilizando o banco de imagens MMU.</i>	90
5.4	<i>Tabulações das curvas ROC resultantes dos experimentos utilizando o banco de imagens MMU.</i>	90

6.1	<i>Definição dos parâmetros da etapa de treinamento.</i>	136
6.2	<i>Definição dos parâmetros da etapa de busca.</i>	136
6.3	<i>Resumo dos resultados da aplicação do algoritmo do ASM.</i>	137

Lista de Códigos

4.2.1 <i>Procedimento da busca local.</i>	63
6.3.1 <i>Rotina para minimizar os efeitos do reflexo especular na pupila.</i> . .	127
A.1.1 <i>Procedimento do Algoritmo Genético.</i>	169
A.2.1 <i>Procedimento do Algoritmo Memético.</i>	173

Simbologia

AG – Algoritmo Genético

AM – Algoritmo Memético

ASM – Active Shape Models

DH – Distância de Hamming

EER – Equal Error Rate

FRR – False Reject Rate (Taxa de falsa rejeição)

FAR – False Accept Rate (Taxa de falsa aceitação)

ROC – Receiver Operating Characteristic

TH – Transformada de Hough

THC – Transformada de Hough Circular

THL – Transformada de Hough Linear

Capítulo 1

INTRODUÇÃO

Na sociedade moderna, está cada vez mais evidente a necessidade de segurança e privacidade. Empresas e entidades públicas ou privadas estão sempre buscando recursos mais eficientes e confiáveis para controlar o acesso de pessoas a lugares e, especialmente, a informações confidenciais. Neste contexto, a biometria surge como uma alternativa interessante para possibilitar a identificação automática de pessoas.

Biometria é a ciência que reconhece um indivíduo através de um sistema eletrônico utilizando a medição de características físicas, biológicas ou comportamentais como, por exemplo, face, impressões digitais, formato da mão, íris, retina, assinatura ou voz.

Normalmente, as pessoas obtém acesso a lugares ou informações seguras usando elementos que elas conhecem, como por exemplo senhas, ou algo que elas carregam consigo, como por exemplo cartões magnéticos ou chaves. Estes recursos não são totalmente confiáveis uma vez que eles podem ser esquecidos, perdidos ou roubados. As técnicas biométricas são baseadas em *quem a pessoa é* ao invés de *o que a pessoa possui* ou *o que a pessoa lembra*,

tornando o sistema mais prático e algumas vezes mais confiável.

Dentre as diversas técnicas biométricas, o reconhecimento de indivíduos pela íris, que é o assunto tratado neste trabalho, chama bastante atenção, uma vez que a íris é um elemento muito rico em detalhes que garantem a unicidade dos padrões de cada indivíduo e possibilitam a construção de sistemas extremamente confiáveis e precisos. A íris é reconhecida como um elemento biométrico há um bom tempo, porém, nos últimos quatro ou cinco anos essa tecnologia tem ganhado uma atenção especial [67]. Isto é comprovado pela enorme quantidade de técnicas de processamento aplicadas ao reconhecimento de íris que têm sido desenvolvidas nesse pequeno período de tempo.

1.1 Objetivos

Os principais objetivos deste trabalho são:

- Analisar e simular um sistema de reconhecimento de íris, utilizando técnicas de processamento tradicionais, aplicado a dois bancos de imagens de íris públicos;
- Fazer uma pesquisa bibliográfica destacando os principais trabalhos da área e as principais técnicas empregadas em cada etapa de processamento de um sistema de reconhecimento de íris;
- Apresentar a proposta de uma nova técnica para localizar a região da íris em uma imagem, baseada em algoritmos meméticos;
- Apresentar a proposta de um método de pré-processamento para amostragem não uniforme de pixels da região da íris, baseado em algoritmos

genéticos, que traz como benefício o aumento da confiabilidade do sistema e/ou a diminuição do tamanho do código gerado e utilizado para comparação; e

- Estudar uma implementação do algoritmo dos Active Shape Models (ASM), apresentá-la e adaptá-la para ser utilizada para segmentar a região da íris com o objetivo de fazer uma primeira avaliação de sua eficiência, identificar as principais dificuldades e desafios desta aplicação e sugerir o desenvolvimento de futuros trabalhos a respeito deste assunto.

A fim de atingir estes objetivos, foram estabelecidas as seguintes metas:

- Estudar, compreender e realizar diversas simulações de cada uma das etapas de processamento da imagem da íris;
- Fazer uma pesquisa bibliográfica profunda em busca dos trabalhos de maior destaque na literatura científica;
- Idealizar um novo método de localização da íris baseado em algoritmos meméticos;
- Estudar a teoria de algoritmos evolutivos, em especial, algoritmos genéticos e algoritmos meméticos que são utilizados como ferramentas nos dois métodos propostos neste trabalho;
- Implementar, em MatLab, o método proposto de localização da íris e realizar todos os testes e simulações necessários para avaliar a eficiência do método;
- Idealizar um novo método para amostrar os pixels da região da íris de maneira não uniforme, implementar o novo método em MatLab e realizar os testes e simulações necessárias para avaliar o método;

- Estudar a teoria do ASM tendo como foco sua possível utilização para segmentar a região da íris;
- Entender os principais detalhes de um algoritmo do ASM implementado em MatLab e fazer as devidas modificações, adaptações e ajustes de parâmetros para utilizá-lo na segmentação da região da íris; e
- Realizar testes e simulações do algoritmo do ASM e interpretar e avaliar os resultados obtidos.

1.2 Estrutura da Tese

Estruturalmente, esta tese está dividida da seguinte forma:

- *Capítulo 1*: Introdução do assunto abordado neste trabalho e apresentação dos objetivos e da estrutura desta tese;
- *Capítulo 2*: Apresentação de alguns conceitos e definições envolvidas no estudo de sistema de reconhecimento de íris, descrição de cada uma das etapas de processamento da imagem da íris e apresentação do estado da arte dos sistemas de reconhecimento de íris;
- *Capítulo 3*: Descrição das implementações de cada etapa de processamento utilizando técnicas tradicionais, apresentação dos bancos de imagens de íris utilizados neste trabalho e apresentação dos resultados obtidos com a simulação do sistema tradicional;
- *Capítulo 4*: Apresentação do método proposto para localizar a região da íris, detalhamento do algoritmo utilizado para sua implementação e, finalmente, exposição dos resultados experimentais e da comparação entre os algoritmos;

- *Capítulo 5*: Apresentação do método proposto para amostragem não uniforme dos pixels da região da íris, explicação dos detalhes envolvidos na sua implementação e exposição dos resultados obtidos a partir da simulação do sistema com dois bancos de imagens diferentes;
- *Capítulo 6*: Justificativa da tentativa de se utilizar ASM para segmentar a região da íris, explicação de todos os detalhes da implementação do algoritmo do ASM, apresentação do pré-processamento aplicado às imagens de íris antes de serem submetidas ao ASM e, finalmente, exposição e análise dos resultados experimentais obtidos;
- *Capítulo 7*: Resumo das implementações e dos resultados obtidos, descrição das principais conclusões e contribuições e sugestão de trabalhos futuros.

Capítulo 2

SISTEMA DE RECONHECIMENTO DE ÍRIS

Os sistemas de reconhecimento biométrico envolvem basicamente duas fases. Primeiro, o usuário precisa se registrar no sistema, permitindo a coleta do elemento biométrico que pode ser a impressão digital, a imagem da íris ou da face, gravação da voz, entre outros elementos mensuráveis. As características-chave são extraídas e convertidas em um código padrão ou *template*, que é armazenado. Na prática, o sistema não grava a foto do olho ou da impressão digital, mas sim o código (*template*) que representa a identidade biométrica do usuário. Em uma segunda fase, para que um usuário tenha acesso ao sistema, é preciso que ele apresente sua característica biométrica, que será comparada ao padrão que foi registrado no banco de dados. A coincidência entre o padrão gravado e o coletado em tempo real

raramente será perfeita. O sistema pode ser configurado para ser mais ou menos tolerante, procurando minimizar o número de rejeições e ao mesmo tempo tentando impedir que um falso usuário obtenha acesso.

As técnicas de reconhecimento por meio da biometria podem ser adotadas para a verificação ou para a identificação [76]. No primeiro caso, o usuário se apresenta como sendo uma determinada pessoa e o sistema confere a veracidade da informação. Se for o caso, o usuário terá acesso ao que é restrito. Sistemas desse tipo são chamados de 1-1 (um-para-um), pois a medida biométrica que se apresenta é simplesmente verificada com a que foi registrada no banco de dados, durante o cadastramento desse usuário.

No segundo caso, a identificação de uma pessoa ocorre quando se tem o dado biométrico dela e se faz uma busca em um banco de dados, comparando as informações até que se encontre (ou não) um registro idêntico ao que é procurado, com certa margem de erro inclusa. Sistemas desse tipo são conhecidos por 1-n (um-para-muitos), pois o dado de uma pessoa é comparado ao de várias outras.

O esquema da Figura 2.1 mostra o processo envolvido em um sistema biométrico de verificação.

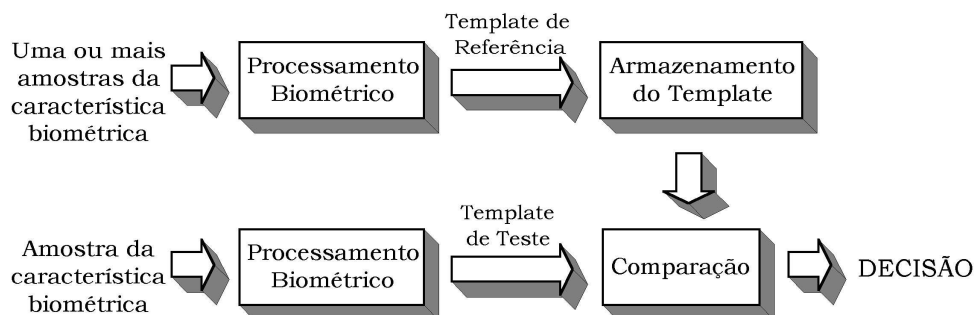


Figura 2.1: Diagrama esquemático do sistema de verificação biométrica.

2.1 Etapas de Processamento

Um sistema de reconhecimento de íris prevê a realização do processamento da imagem do olho em diversas etapas [17, 79], que são: a localização da região da íris na imagem de olho; a normalização da região da íris para lidar com as inconsistências dimensionais; o processo de extração e codificação de características e a etapa de comparação. O esquema da Figura 2.2 mostra uma visão geral do sistema de reconhecimento de íris.

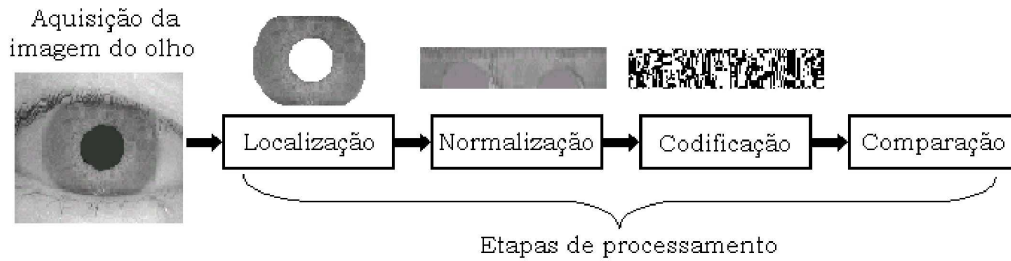


Figura 2.2: *Etapas de processamento do sistema de reconhecimento de íris.*

Comentários a respeito de cada uma das etapas de processamento serão apresentados a seguir.

2.1.1 Etapa de localização

No momento da aquisição, ainda não é possível obter uma imagem contendo somente a íris. Na verdade, a íris representa uma parte de uma imagem que também contém as áreas ao redor do olho, como mostra a imagem da Figura 2.3. Portanto, o objetivo da primeira etapa de processamento de um sistema de reconhecimento de íris é localizar e isolar a porção da imagem que corresponde à íris.

A borda da pupila e a borda externa da íris, que juntas definem a região da íris, apresentam um contorno aproximadamente elíptico. Contudo, as



Figura 2.3: *Imagem de um olho do banco de imagens UBIRIS [63].*

pesquisas mais tradicionais na área de reconhecimento de íris, com o objetivo de simplificar o processamento, aproximam estas bordas a círculos com perdas mínimas no desempenho do sistema. Assim, geralmente, o processo de localização da íris envolve a aplicação de algoritmos de detecção de círculos.

Quando as pálpebras ou os cílios ocultam uma parte da íris, somente a porção da imagem abaixo da pálpebra superior e acima da pálpebra inferior deve ser utilizada. Geralmente, existe um contraste dos cílios e das bordas da pálpebra com relação à íris, facilitando a detecção desses elementos.

A eficiência da etapa de localização é muito importante para o sucesso do sistema de reconhecimento de íris, pois, uma falsa representação de algumas regiões como pertencentes à região da íris irá corromper o *template* biométrico provocando erros de reconhecimento.

Em geral, é possível identificar duas estratégias principais nas quais as diversas metodologias de localização da íris são baseadas. Uma das estratégias é baseada em padrões e geralmente consiste na maximização de algumas equações, como é o caso dos trabalhos de Daugman [16], Kim et al. [39] e Poursaberi [62]. A outra estratégia é baseada em fronteiras e exige a geração

de um mapa de bordas para ser utilizado por algum algoritmo de interpolação geométrica.

O método baseado em padrões mais tradicional é o operador integro-diferencial proposto por Daugman [16]. Os métodos mais importantes baseados em fronteiras utilizam a Transformada de Hough Circular (THC) para encontrar círculos em um mapa de bordas. A utilização desta transformada foi primeiramente sugerida por Wildes [78].

Neste trabalho foram utilizadas implementações destes dois métodos de segmentação. Para detecção das pálpebras e dos cílios foi utilizado um método baseado na Transformada de Hough Linear proposto por Libor Masek [50]. Na seção 3.1 serão apresentados os detalhes destas implementações.

No capítulo 4 será proposto um novo método de localização baseado em fronteira que utiliza uma aplicação de Algoritmos Meméticos (AM).

2.1.2 Etapa de normalização

Uma vez que a região referente a íris foi devidamente localizada em uma determinada imagem, tem-se a necessidade de organizar as informações contidas nessa região de forma a permitir futuras comparações. Para que estas comparações se processem com a devida precisão é essencial que as imagens possuam uma dimensão constante. Porém, alguns fatores contribuem para uma inconsistência dimensional dessas imagens como a variação da distância entre o eixo ótico da câmara e o olho e também a variação do tamanho da pupila causada por diferentes níveis de luminosidade. Além disso, ainda existem outros fatores que agregam inconsistências entre tais imagens como uma possível rotação da câmara, inclinação da cabeça ou movimentação do olho dentro do globo ocular [50]. É importante lembrar que a translação da íris com relação à câmara não representa problema, pois a íris será localizada

na imagem da mesma maneira.

Outra questão a se considerar é o fato de que as regiões da pupila e da íris não são sempre concêntricas. Geralmente a pupila apresenta uma posição levemente nasal, ou seja, seu centro está em uma posição inferior ao centro da íris e mais próxima do nariz [4].

O processo de normalização é responsável por gerar imagens com dimensões constantes, assim, imagens da mesma íris capturadas sob condições diferentes terão suas características em uma mesma localização espacial.

O método mais tradicional e amplamente utilizado nessa etapa é o método proposto por John Daugman [15, 16, 17, 18] que faz uma representação da região anelar da íris utilizando coordenadas polares adimensionais gerando uma representação retangular de tamanho fixo. Este processo envolve uma amostragem uniforme de pixels na região da íris.

Neste trabalho foi utilizada a técnica de normalização proposta por John Daugman, que foi implementada por Libor Masek [50]. Os detalhes dessa implementação são apresentados na seção 3.2 do capítulo 3 deste trabalho.

No capítulo 5 é proposta uma técnica de pré-processamento baseada em Algoritmos Genéticos (AG) em que o processo de amostragem dos pixels, que ocorre na etapa de normalização, é realizado de maneira não uniforme com o objetivo de representar as características da íris de forma mais eficiente, possibilitando gerar templates menores sem prejudicar o desempenho do sistema de reconhecimento de íris.

2.1.3 Etapa de extração e codificação de características

Nesta etapa é realizada, inicialmente, a extração das características da íris e, em seguida, essas características são codificadas.

No processo de extração de características, os detalhes mais significativos

da estrutura da íris, que a caracterizam, devem ser detectados na imagem normalizada da íris. Depois, as características extraídas devem ser codificadas para gerar um *template* que será utilizado para comparação.

Existem diversos métodos utilizados para extração de características. Em geral, esses métodos podem ser agrupados em três categorias: métodos baseados em fase, representações de cruzamento por zero e métodos baseados em análise de textura.

Como exemplos de métodos baseados em fase pode-se citar os filtros de Gabor 2D que foram empregados em diversos trabalhos [15, 17, 18] e ainda, os filtros Log-Gabor aplicados por Vasta et al. [75] e por Libor Masek [50]. O melhor exemplo de representação de cruzamento por zero é o trabalho de Boles and Boashash [7] que calculou a representação de cruzamento por zero da Transformada Wavelet 1D em vários níveis de resolução. Os métodos baseados em análise de textura utilizam o Laplaciano de filtros Gaussianos, como é o caso do trabalho de Wildes et al. [78] e o Momento Hermite Gaussiano que foi usado por Li Ma et al. [47] para caracterizar variações locais da intensidade do sinal. Muitos outros trabalhos apresentam métodos de extração de características baseados em análise de textura [6, 39, 43, 46, 45, 80].

Uma vez que as características da íris são extraídas, deve ser aplicado algum algoritmo de codificação para gerar um *template*. Na maioria das vezes este *template* gerado é binário, porém, ele também pode ser representado por um vetor de características composto de números inteiros ou reais.

Neste trabalho foi utilizada uma variação do filtro de Gabor conhecida como filtro Log-Gabor que foi proposta por Field [30]. Depois, foi aplicado um algoritmo de quantização para gerar *templates* binários. Os detalhes da implementação desses métodos são apresentados na seção 3.3.

2.1.4 Etapa de comparação

Uma vez que o *template* representativo da região da íris foi gerado na fase de codificação, deve-se agora utilizar uma métrica para comparar os *templates* gerados a partir de duas imagens de íris. Esta métrica deve ser capaz de medir o grau de similaridade de dois *templates* pela obtenção de uma faixa de valores quando são comparados *templates* de imagens da mesma íris (comparação intra-classe) e outra faixa de valores quando são comparados *templates* de íris diferentes (comparação inter-classes).

Para que o processo seja eficiente e apresente um alto grau de confiabilidade, essas faixas de valores obtidos devem ser bem distintas entre si para permitir a correta decisão, ou seja, se os *templates* foram gerados à partir da mesma íris ou não.

Diversas métricas podem ser utilizadas para comparação sendo que as mais populares são a distância de Hamming (dH), a distância euclidiana ponderada e a correlação normalizada.

A distância de Hamming pode ser utilizada quando os *templates* são binários. Ela fornece uma medida da quantidade de bits que são comuns entre dois *templates* comparados. Vários trabalhos utilizam essa métrica como, por exemplo, os de John Daugman [15, 16, 17, 18], de Vasta et al. [75] e de Li Ma et al. [46].

A distância euclidiana ponderada pode ser utilizada quando os *templates* são compostos de valores inteiros. Ela fornece uma medida da similaridade de um conjunto de valores provenientes de dois *templates*. Esta técnica foi empregada por Zhu et al. [80].

A correlação normalizada foi utilizada como métrica para comparação de dois *templates* por Wildes et al. [78] e por Kim et al. [39].

Neste trabalho foi utilizada a distância de Hamming como métrica para

comparação. Os detalhes de sua aplicação são apresentados na seção 3.4.

2.2 Ambiente de Decisão e as Taxas de Erro

Como visto, um sistema de reconhecimento de íris pode utilizar diversas métricas para comparar dois *templates*. Para auxiliar na definição de “ambiente de decisão” e das taxas de erro, suponha a utilização da distância de Hamming (dH) como métrica, lembrando que um raciocínio similar se aplica a qualquer outra métrica.

Quando são processadas inúmeras comparações intra-classe ou inter-classes, os valores de distância de Hamming obtidos podem ser aproximados a uma distribuição normal. O grande objetivo de um sistema de reconhecimento de íris é ser capaz de obter uma boa separação entre as distribuições de distância de Hamming geradas a partir de comparações intra-classe e inter-classes. A representação das duas distribuições é chamada de “ambiente de decisão”, pois ela revela o quanto os dois casos estão separados e conseqüentemente, o quanto o sistema é confiável. A Figura 2.4 mostra um exemplo de ambiente de decisão.

Para a geração da distribuição intra-classe são efetuadas todas as comparações possíveis entre imagens da mesma íris e são calculados os respectivos valores de dH. Em seguida, conta-se o número de ocorrência de valores de dH em determinadas faixas de valores pré-estabelecidos e plota-se a curva “Quantidade de ocorrências” versus “Faixa de valores de dH”. Para a geração da distribuição inter-classes é utilizado o mesmo procedimento, porém são efetuadas comparações entre íris diferentes.

Para decidir se dois *templates* foram gerados à partir de uma mesma íris ou não, deve-se pré-determinar um valor de distância de Hamming para ser o

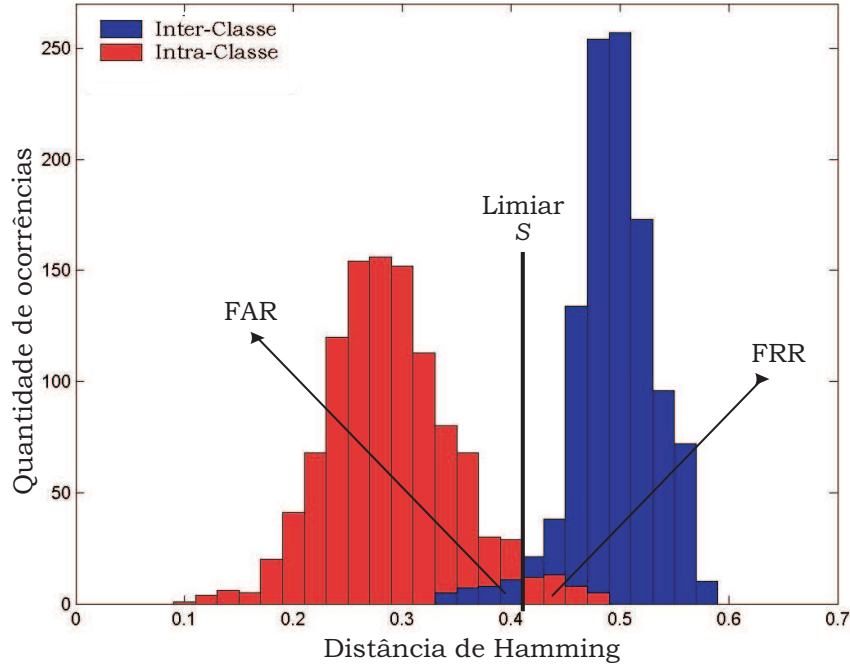


Figura 2.4: *Ambiente de decisão: distribuições de distância de Hamming geradas a partir de comparações intra-classe e inter-classes.*

ponto de separação (limiar) entre as duas distribuições. Porém, os sistemas de biometria, em geral, não são perfeitos. Pode ocorrer uma sobreposição entre as duas distribuições, o que representará erros na tomada de decisão.

Nos sistemas de biometria são tratados dois tipos de taxas de erro. A primeira é chamada de taxa de falsa aceitação (FAR - False Accept Rate) que representa a probabilidade de um impostor ser aceito pelo sistema. A outra taxa é chamada de taxa de falsa rejeição (FRR - False Reject Rate) que representa a probabilidade de um indivíduo apto ser considerado impostor e, portanto, ser recusado pelo sistema. Essas taxas são calculadas à partir da sobreposição entre as distribuições intra e inter-classes. A FAR é obtida pela área da distribuição inter-classe D_{inter} entre 0 e o limiar s dividida pela área total dessa distribuição (Equação (2.1)). Na prática isto representa a

quantidade de valores de dH ocorridos entre 0 e o limiar s na distribuição inter-classes dividida pela quantidade total de comparações inter-classes efetuadas. A FRR é obtida pela área da distribuição intra-classe D_{intra} entre o limiar s e 1 dividida pela área total dessa distribuição (Equação (2.2)). Na prática isto representa a quantidade de valores de dH ocorridas entre o limiar s e 1 na distribuição intra-classe dividida pela quantidade total de comparações intra-classe efetuadas.

$$FAR = \frac{\int_0^s D_{inter}(x)dx}{\int_0^1 D_{inter}(x)dx} \quad (2.1)$$

$$FRR = \frac{\int_s^1 D_{intra}(x)dx}{\int_0^1 D_{intra}(x)dx} \quad (2.2)$$

É fácil perceber que o valor do limiar influencia bastante nas taxas de erro. Quando se aumenta o valor do limiar, a FAR aumenta e a FRR diminui, enquanto quando se diminui o valor do limiar, a FAR diminui e a FRR aumenta. Neste trabalho, ao escolher o valor do limiar de um sistema, o objetivo é a minimização da soma das duas taxas.

Na prática, para garantir a segurança de um sistema de modo que nenhum impostor seja aceito, geralmente, procura-se diminuir bastante a FAR, mesmo que isso represente um aumento na FRR.

2.3 Estado da Arte

A idéia de utilizar a íris como elemento de identificação pessoal foi sugerida pela primeira vez pelos oftalmologistas Flom e Safir em 1987 [32]. Em 1993, John Daugman [16, 18] apresentou o primeiro e mais conhecido algoritmo para reconhecimento automático de indivíduos utilizando a estrutura da íris. Para executar a etapa de segmentação, Daugman aplicou um operador integro-diferencial para encontrar os círculos interno e externo que definem a região da íris. Uma vez que a íris está localizada, o processo de normalização é executado convertendo-se a imagem da região da íris para a forma cartesiana, projetando-a em um sistema de coordenadas polares adimensionais. Este procedimento converte a região anelar da íris para uma representação retangular. O algoritmo de extração de características aplica filtros complexos 2D Gabor Wavelets e um *template* binário é gerado pela codificação das informações de fase dos coeficientes produzidos. A comparação das íris é realizada utilizando-se a análise do valor de distância de Hamming (dH) calculado a partir de dois *templates*. Se a dH for menor que um determinado valor de limiar, os *templates* são considerados originados da íris do mesmo indivíduo, caso contrário, eles são de indivíduos diferentes.

Depois do sistema de Daugman, muitos outros algoritmos foram propostos para processar cada etapa do sistema de reconhecimento de íris. Wildes [78, 79] desenvolveu outro algoritmo bem conhecido. Para localizar a região da íris, a Transformada de Hough (TH) [34, 38, 51] foi aplicada em um mapa de bordas binário que foi originado utilizando-se um algoritmo de detecção de bordas baseado em informações de gradiente, com o objetivo de encontrar os círculos que definem a região anelar da íris. As pálpebras superior e inferior são detectadas usando a Transformada de Hough parabólica. Esta metodologia de localização da região da íris também foi utilizada, com diver-

sas pequenas variações, em muitos trabalhos como por exemplo [40] e [50]. O algoritmo de Wildes realiza um tipo de normalização em que a imagem é deformada geometricamente por sucessivos deslocamentos e rotações, até que um melhor ajuste a outra imagem seja encontrado pela minimização da distância quadrática média entre as imagens. Para codificar as características da íris, foi construída uma pirâmide Laplaciana com quatro diferentes níveis de resolução. Para a comparação é aplicada a correlação normalizada e a discriminação linear de Fisher.

Boles e Boashash [7] utilizaram um algoritmo baseado em detecção de bordas para processar a localização da íris. Eles calcularam o cruzamento por zero da transformada wavelet diádica de uma dimensão em vários níveis de resolução, sobre círculos concêntricos na imagem da íris. A etapa de normalização é realizada aplicando-se escalas às imagens de forma que o diâmetro da íris tenha um tamanho fixo e comum, assegurando uma correspondência entre os círculos virtuais que serão determinados em diferentes imagens. Um sinal unidimensional é produzido e comparado às características modelo utilizando-se duas funções dessemelhantes, sendo que uma delas é gerada usando todos os pontos da representação e a outra usando somente os pontos de cruzamento por zero. As principais vantagens deste algoritmo é que ele não é influenciado por variações de translação, rotação e escala, assim como, é relativamente imune a condições de ruído e mudanças de iluminação.

Tisse, et al. [74] também apresentaram algumas inovações. Para a localização da íris eles utilizaram uma combinação da Transformada de Hough e do operador integro-diferencial, obtendo a vantagem de evitar erros devido à reflexões especulares. Eles introduziram a idéia de frequência emergente e fase instantânea e propuseram um algoritmo eficiente computacionalmente para a etapa de extração e codificação das características. A transformada Hilbert é utilizada para criar uma imagem analítica e o código da íris (*template*)

é gerado por um processo de limiarização dos modelos de frequência emergente e a parte real e imaginária da fase instantânea. Para a normalização e a etapa de comparação, foram utilizadas as técnicas sugeridas por Daugman, isto é, uma imagem normalizada foi criada usando coordenadas polares adimensionais e a distância de Hamming foi considerada para comparar dois templates.

Lim, et al. [43] propuseram novos métodos somente para processar a extração de características e a etapa de comparação. Eles decompuseram a imagem da íris em quatro níveis usando a Transformada Wavelet Haar 2D e quantizaram as informações de alta frequência do quarto nível para formar um código binário de 87 bits. O algoritmo de comparação utiliza uma rede neural de aprendizagem competitiva (LVQ) modificada que é otimizada por um minucioso processo de seleção dos vetores de peso iniciais.

Dargham et. al. [14] processaram a etapa de localização utilizando um método relativamente simples baseado em limiarização de imagem, com o objetivo de obter os parâmetros dos dois círculos que definem a região da íris, pela maximização de uma função. A região da íris segmentada é então reconstruída em um formato retangular. Finalmente, redes neurais auto-organizáveis são utilizadas para fazer a comparação.

Wang e outros apresentaram alguns algoritmos para localizar a região da íris em [13, 37, 46]. Todos eles são baseados na Transformada de Hough com diversas pequenas variações. Juntamente com Ma e outros, Wang também propôs três diferentes algoritmos para as etapas de extração de características e comparação. Em [48] adotaram filtros de Gabor multi-canais para a extração de características e utilizou a distância Euclidiana ponderada para a comparação. Em outro trabalho [44] utilizaram filtros circulares simétricos para capturar informações locais de textura da íris que são usadas para construir um vetor de características de comprimento fixo. Para comparação

foi utilizado um método modificado de linha de característica mais próxima. Finalmente, em [45] conseguiram uma extração de características eficiente construindo um banco de filtros espaciais. A dimensão do *template* é reduzida pela aplicação do discriminante linear de Fisher e, então, o classificador de centro mais próximo foi adotado para classificação.

Zhu et al. [80] empregaram o filtro de Gabor multi-canal e a transformada wavelet 2D para capturar os detalhes globais e locais da íris. Para identificação foi utilizada a classificação pela distância euclidiana ponderada. Noh et al. [55] propuseram um novo algoritmo para extrair as características da íris utilizando Multi-resolution Independent Component Identification (M-ICA). Chen e Yuan [9] particionaram a imagem da íris em pequenos blocos e computaram as características dimensionais fractais locais para construir o *template* da íris. Então, os *templates* são comparados utilizando-se a k-médias e redes neurais.

Du et al. [24] também propuseram um método diferente para realizar a extração de características e a comparação. Uma escala de cinza invariante chamada Modelos de Textura Local (Local Texture Patterns - LTP) foi utilizada para comparar a intensidade de um pixel com a intensidade média dos pixels da sua vizinhança. A informação LTP é projetada de 2D para 1D e o vetor resultante é normalizado de forma que a soma de seus elementos seja igual a 1. Para realizar a comparação, eles utilizaram um valor derivado do produto de duas medidas, sendo uma delas baseada na tangente do ângulo entre os dois vetores comparados, e a outra, baseada na entropia relativa de um vetor com respeito ao outro.

A literatura científica apresenta várias técnicas para localização de íris. Al-Zubi e Abu-Al-Nadi [5] aplicaram a Transformada de Hough juntamente com uma técnica de ajuste polinomial e alguns operadores morfológicos para segmentar a íris. Kim et. al. [39] consideraram a hipótese de que os valores

de nível de cinza das imagens podiam ser representados por uma composição de três distribuições gaussianas. Assim, eles utilizaram um algoritmo de maximização de expectativa [22] para estimar os parâmetros da respectiva distribuição. A intenção era que uma das distribuições representasse os pixels mais escuros (região da pupila), enquanto outra, os pixels com intensidade intermediária (região da íris) e a terceira, os pixels mais claros (áreas de reflexos).

Du et al. [25] propuseram um método que consistia em, primeiramente, detectar a região da pupila, depois, representar a imagem por coordenadas polares e, finalmente, localizar a borda externa da íris considerando que ela é a maior borda horizontal encontrada pela aplicação do filtro de Sobel. Mira e Mayer [52] aplicaram operadores morfológicos para detectar, primeiramente, a borda da pupila e depois a borda externa da íris.

Watson [77] propôs um algoritmo para localização rápida da íris. A detecção da borda da pupila é realizada procurando o centro de massa de uma superfície homogênea (uma característica da região da pupila), e depois, um determinado número de pixels da íris são coletados da região em torno da pupila e armazenados para um futuro processamento. Proença e Alexandre [64] desenvolveram um algoritmo capaz de localizar a região da íris em imagens com ruídos. Para obterem robustez, eles combinaram o método empregado por Wildes [79] com um processo de formação de “clusters”.

Poursaberi e Araabi [62] utilizaram operadores morfológicos para a detecção da pupila e depois, selecionaram um determinado raio apropriado em torno da pupila para abranger a região da íris que contém o colarete. He e Shi [36] propuseram um método geométrico para a detecção da pupila. A borda externa da íris foi localizada utilizando-se a Transformada de Hough que foi aplicada a uma imagem reduzida com o objetivo de diminuir o processamento computacional.

2.4 Considerações Finais do Capítulo

Neste capítulo foram abordadas as principais características dos sistemas biométricos de reconhecimento de íris, com ênfase na definição das principais etapas de processamento que estão envolvidas nestes sistemas. A importância e a finalidade de cada etapa de processamento foram apresentadas na seção 2.1.

Na seção 2.2 foram definidas as taxas de erro e o termo “Ambiente de decisão” que são utilizados para avaliar a eficiência do sistema de reconhecimento.

Finalmente, na seção 2.3 foi apresentado o estado da arte das pesquisas e das técnicas utilizadas em sistemas de reconhecimento de íris.

Capítulo 3

SISTEMA DE RECONHECIMENTO DE ÍRIS UTILIZADO PARA SIMULAÇÃO

Neste trabalho foi implementado em MatLab um sistema de reconhecimento de íris utilizando as técnicas tradicionais em cada etapa de processamento. Grande parte das implementações em MatLab que são discutidas neste capítulo foram disponibilizadas por Libor Masek [50].

Nas próximas seções são descritos os métodos que foram implementados para processar cada etapa do sistema. Em seguida, são apresentados os bancos de imagens de íris utilizados neste trabalho e ainda, os resultados obtidos com a simulação do sistema implementado.

3.1 Métodos de Localização

Foram utilizadas para simulação, as implementações dos dois métodos mais tradicionais de localização da região da íris que aproximam as bordas da íris a círculos, a Transformada Circular de Hough e o Operador Integro-Diferencial.

Para detectar as pálpebras e os cílios, foi utilizado o método proposto por Libor Masek [50].

Os dois métodos de localização da íris e o método de detecção das interferências são descritos a seguir.

3.1.1 Transformada de Hough circular

A Transformada de Hough (TH) [34, 38, 51] foi proposta por Paul Hough em 1962 e é uma técnica utilizada em aplicações que envolvem a detecção e a determinação dos parâmetros de objetos geométricos simples em imagens digitais, como por exemplo retas, círculos e elipses. A Transformada de Hough Circular (THC) pode ser utilizada para se obter o raio e as coordenadas do centro das bordas circulares da pupila e da parte externa da íris.

Em geral, a TH é aplicada após a imagem passar por um processo de detecção de bordas. Assim, somente os pixels da imagem referentes às bordas são utilizados no processamento. Para fazer a detecção de bordas foi utilizado o método de Canny [8]. Na implementação utilizou-se a função do MatLab desenvolvida por Kovsi [41] que é uma versão modificada do método de Canny e permite adicionar informações de gradiente. O algoritmo de detecção de bordas depende do ajuste de alguns parâmetros de limiar que determinam a detecção de um contraste maior ou menor, influenciando nas características do mapa de bordas gerado.

O procedimento de Hough requer a geração de uma matriz de acumulação de votos com o número de dimensões igual ao número de parâmetros necessários para definir a forma. No caso de círculo, o acumulador tem 3 dimensões, como ilustrado na Figura 3.1.

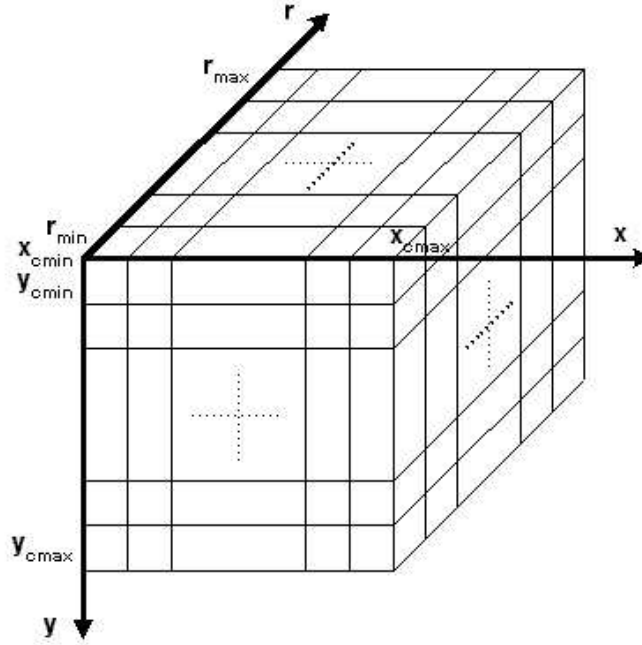


Figura 3.1: *Acumulador de votos.*

Neste trabalho foi utilizada uma matriz de acumulação de votos com dimensão $[n^\circ \text{ de colunas da imagem} \times n^\circ \text{ de linhas da imagem} \times (r_{max} - r_{min})]$. Onde r_{max} e r_{min} representam a faixa dos possíveis valores que o raio do círculo que se deseja encontrar pode assumir.

Deve-se enfatizar que, apesar deste trabalho se referir sempre à uma busca por círculos, na verdade, busca-se apenas o contorno do círculo, ou seja, uma circunferência. Uma circunferência pode ser definida pela equação $(x - x_c)^2 + (y - y_c)^2 = r^2$, onde x_c e y_c são as coordenadas do centro da circunferência (ou do círculo), r é o raio da circunferência (ou do círculo) e x e y são as coordenadas de um ponto pertencente à circunferência (ou ao contorno do

círculo).

Na THC um pixel de borda com coordenadas (x, y) no espaço de imagem é mapeado para o espaço de parâmetros definindo-se dois dos parâmetros (por exemplo, x_c e y_c) e achando-se o terceiro (por exemplo, r) que resolve a equação da circunferência. Desta forma, obtém-se o ponto (x_c, y_c, r) no espaço de parâmetros, o que representa um possível círculo existente na imagem.

Para cada pixel de borda da imagem esse procedimento é repetido para se obter todas as combinações possíveis de (x_c, y_c, r) , ou seja, cada valor de x_c entre x_{cmin} e x_{cmax} é combinado com cada valor de y_c entre y_{cmin} e y_{cmax} para se obter o valor de r correspondente.

A cada tripla (x_c, y_c, r) obtida, é incrementado o valor do acumulador nessa posição $A(x_c, y_c, r)$, ou seja, é atribuído um voto a essa posição.

Quando todos os pixels tiverem sido processados, é procurado no acumulador A os maiores valores, ou seja, as posições que receberam mais votos, que indicaram os parâmetros de prováveis círculos na imagem.

O raio da borda externa da íris e o da pupila possuem tamanho médio bem diferentes, portanto, a detecção dessas duas circunferências é realizada separadamente, ou seja, deve-se aplicar todo o algoritmo para se detectar o círculo da íris e depois repeti-lo para a pupila. A detecção da íris é realizada primeiro para que na detecção da pupila seja utilizada somente a região interior à íris ao invés da imagem inteira. Este procedimento diminui a quantidade de processamento.

O método da TH apresenta alguns problemas. O primeiro que se pode citar é a necessidade de um pré-processamento de detecção de bordas que requer a escolha de valores de limiar. Esta escolha pode resultar na remoção de pontos críticos para a correta detecção de círculos. Outra desvantagem é o fato de precisar de um processamento computacional intenso, não sendo um método adequado para aplicações em tempo real.

O processamento computacional aumenta com o acréscimo no tamanho da matriz de acumulação de votos e também com o aumento da quantidade de pixels de borda da imagem. A resolução do vetor de acumulação determina a exatidão com que os parâmetros podem ser determinados.

Foi utilizada a sugestão de Wildes et al. [79] que utilizaram informações de gradiente para a geração do mapa de bordas. Apenas o gradiente vertical foi utilizado para se detectar a borda externa da íris e o gradiente horizontal para se detectar as pálpebras. Para a detecção da borda da pupila foram utilizados os gradientes vertical e horizontal, ou seja, o mapa de bordas completo. A Figura 3.2 ilustra os diferentes mapas de bordas gerados quando se variam as informações de gradiente.

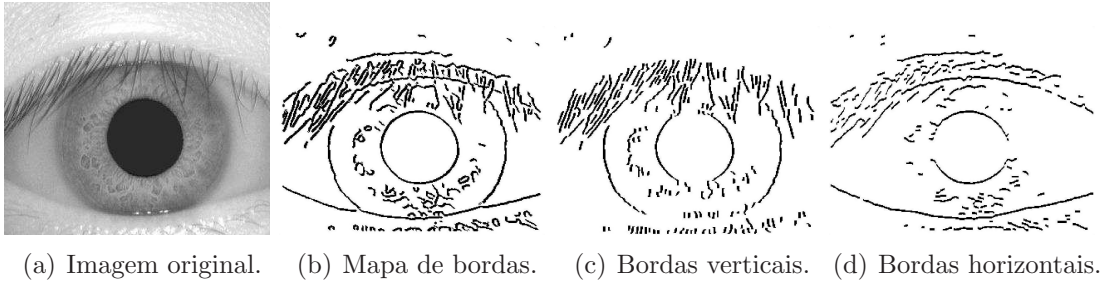


Figura 3.2: a)Imagem de um olho; b)Mapa de bordas correspondente; c)Mapa de bordas utilizando somente o gradiente vertical; d)Mapa de bordas utilizando somente o gradiente horizontal.

3.1.2 Operador integro-diferencial

O Operador Integro-Diferencial foi proposto por John Daugman [16] e tem o objetivo de localizar as bordas circulares da pupila e da parte externa da íris. A Equação (3.1) define esse operador.

$$\max(r, x_0, y_0) \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r, x_0, y_0} \frac{I(x, y)}{2\pi r} ds \right| \quad (3.1)$$

onde $I(x, y)$ representa a intensidade dos pixels da imagem do olho, r é o

raio a ser procurado, $G_\sigma(r)$ representa uma função gaussiana de suavização e s é o contorno do círculo de raio r e centro nas coordenadas (x_0, y_0) .

O operador procura pelo caminho circular onde existe uma maior mudança nos valores dos pixels quando ocorre uma variação no raio e nas coordenadas (x, y) do centro do círculo. O operador é aplicado ao mesmo tempo que o grau de suavização é reduzido progressivamente com o objetivo de conseguir uma localização precisa.

Como não se pode assumir que o contorno circular externo da íris tem o mesmo centro do contorno circular da pupila, os três parâmetros que definem o círculo externo da íris devem ser estimados separadamente aos que definem o círculo da pupila.

O operador integro-diferencial pode ser visto como uma variação da Transformada de Hough [50] uma vez que ele também faz uso da primeira derivada da imagem e executa uma busca por parâmetros geométricos. Como o operador trabalha com as informações brutas da derivada, ele não enfrenta os problemas de ajuste de limiares que acontecem com a Transformada de Hough. Entretanto, o algoritmo pode falhar onde existirem ruídos na imagem, provenientes, por exemplo, de reflexos.

Neste trabalho foi utilizada parte da implementação do operador integro-diferencial desenvolvida e disponibilizada por Anirudh S. [68].

3.1.3 Detecção das pálpebras e dos cílios

Foi utilizado o procedimento sugerido por Libor Masek [50] para detectar as regiões da íris corrompidas pelas pálpebras e pelos cílios. Para se isolar as pálpebras admitiu-se que a borda da mesma pode ser aproximada por um segmento de reta. O primeiro passo é encontrar uma reta correspondente à borda da pálpebra superior e uma correspondente à borda da pálpebra

inferior. Para isto foi utilizada a Transformada de Hough Linear. Uma segunda reta é então desenhada horizontalmente interceptando a primeira no ponto de borda da íris mais próximo à pupila. Este procedimento é realizado tanto para a pálpebra superior quanto para a inferior e são excluídas as regiões acima da reta horizontal referente à pálpebra superior e abaixo da referente à pálpebra inferior. Este processo está ilustrado na Figura 3.3. Ao se traçar a segunda reta, garante-se uma isolamento máxima da região pertencente às pálpebras. Para a detecção das bordas utilizou-se o método de Canny e somente as informações de gradiente horizontal [50].

Pode acontecer de, em algumas imagens, não existir a oclusão da íris pelas pálpebras. Assim, se o máximo valor no espaço de Hough for menor do que um limiar pré-estabelecido, então, nenhuma reta é identificada, o que representa uma não oclusão. Além disso, uma reta só é considerada quando ela se encontra fora da região da pupila e dentro da região da íris [50].

Para se isolar os cílios é utilizada uma técnica baseada no estabelecimento de um limiar, uma vez que, para o conjunto de imagens utilizado, os cílios são sempre mais escuros quando comparados com o resto da imagem. Assim, todos os pixels da imagem com tom de cinza mais escuro do que o limiar estabelecido são considerados pixels pertencentes aos cílios e são excluídos.

Os pixels pertencentes às pálpebras e aos cílios que foram detectados durante o processo são marcados utilizando-se o tipo NaN (not a number) do MatLab para que, nas próximas etapas, sejam identificados como interferências na região da íris.

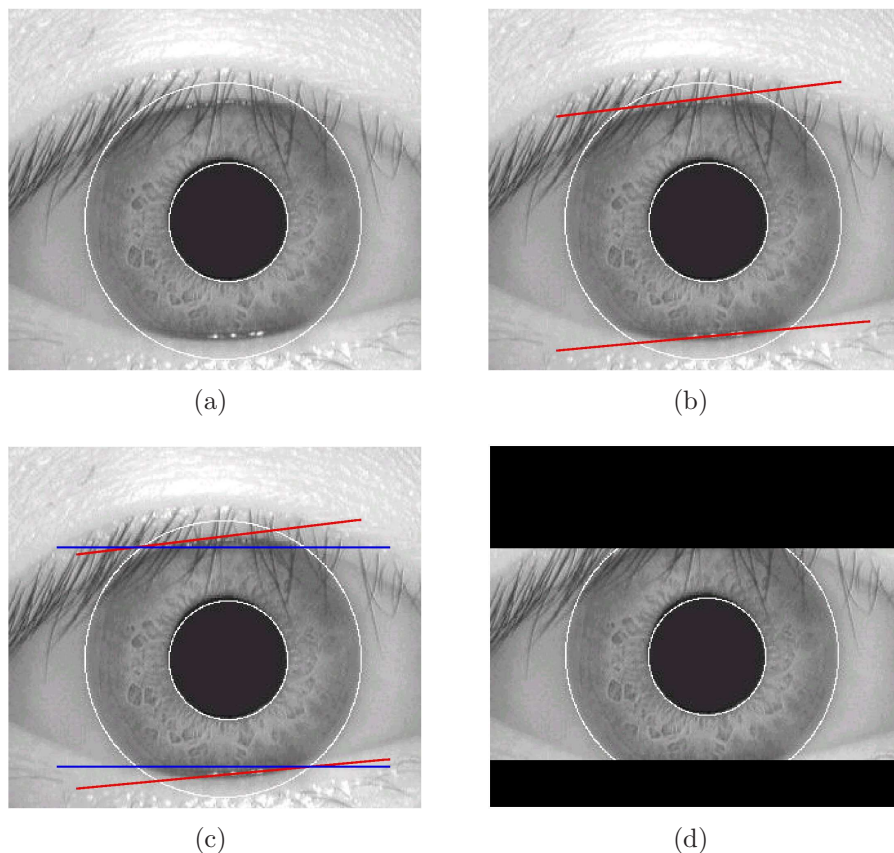


Figura 3.3: *Estágios do processo de exclusão das pálpebras. a) Imagem original proveniente do banco de imagens CASIA [1]; b) Retas detectadas utilizando-se a Transformada de Hough Linear correspondente às bordas das pálpebras superior e inferior; c) Retas horizontais que interceptam as primeiras no ponto de borda da íris mais próximo à pupila; d) Exclusão das regiões acima da reta horizontal referente à pálpebra superior e abaixo da referente à pálpebra inferior.*

3.2 Método de Normalização

John Daugman propôs um método para gerar uma representação retangular da região anular da íris utilizando-se coordenadas polares adimensionais [15, 16, 17, 18]. Esta representação é mostrada na Figura 3.4.

Esse método modela a região da íris como se esta fosse um anel de borracha ancorado na borda externa da íris, tendo o centro da pupila como o ponto de referência.

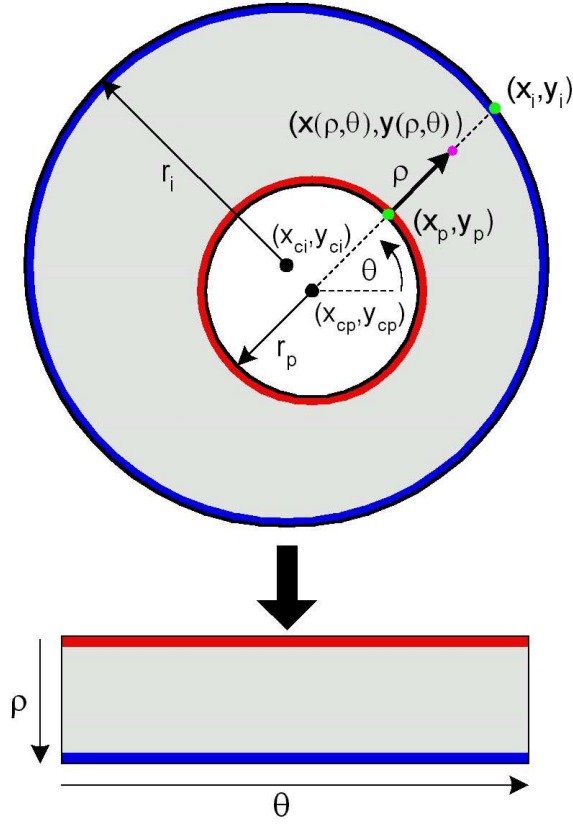


Figura 3.4: *Representação retangular da região da íris proposta por John Daugman [15, 16, 17, 18].*

O método proposto consiste em re-mapear a região da íris passando de coordenadas cartesianas (x, y) para coordenadas polares adimensionais (ρ, θ) onde ρ pertence ao intervalo $[0, 1]$ e θ representa um ângulo no intervalo $[0, 2\pi]$. A representação polar normalizada é modelada utilizando-se a Equação (3.2).

$$I(x(\rho, \theta), y(\rho, \theta)) \rightarrow I(\rho, \theta) \quad (3.2)$$

$$\begin{cases} x(\rho, \theta) = x_p(\theta) + \rho(x_i(\theta) - x_p(\theta)) \\ y(\rho, \theta) = y_p(\theta) + \rho(y_i(\theta) - y_p(\theta)) \end{cases} \Rightarrow \begin{cases} x(\rho, \theta) = (1 - \rho)x_p(\theta) + \rho x_i(\theta) \\ y(\rho, \theta) = (1 - \rho)y_p(\theta) + \rho y_i(\theta) \end{cases} \quad (3.3)$$

com

$$\begin{cases} x_p(\theta) = x_{cp} + r_p \cos \theta \\ y_p(\theta) = y_{cp} + r_p \sin \theta \end{cases} \quad (3.4)$$

$$\begin{cases} x_i(\theta) = x_{ci} + r_i \cos \theta \\ y_i(\theta) = y_{ci} + r_i \sin \theta \end{cases} \quad (3.5)$$

onde $I(x, y)$ é a imagem da região da íris, (x, y) são as coordenadas cartesianas originais, (ρ, θ) são as coordenadas polares normalizadas correspondentes, (x_p, y_p) e (x_i, y_i) são as coordenadas da borda da pupila e da íris na direção θ , enquanto (x_{cp}, y_{cp}) e r_p e (x_{ci}, y_{ci}) e r_i são as coordenadas do centro e o raio da pupila e da íris, respectivamente.

Na prática, para se aplicar o método e gerar a representação retangular deve-se considerar o centro da pupila como ponto de referência e passar vetores radiais pela região da íris, como mostrado na Figura 3.5. Uma determinada quantidade de pontos é selecionada uniformemente ao longo de cada linha radial. Essa quantidade representa a resolução radial que define a dimensão vertical da representação retangular. Já a quantidade de linhas radiais representa a resolução angular que define a dimensão horizontal da representação retangular.

Como as regiões da pupila e da íris podem ser não concêntricas, Libor Masek [50] sugeriu em sua implementação, a utilização da Equação (3.6) para redimensionar os pontos dependendo do valor do ângulo θ .

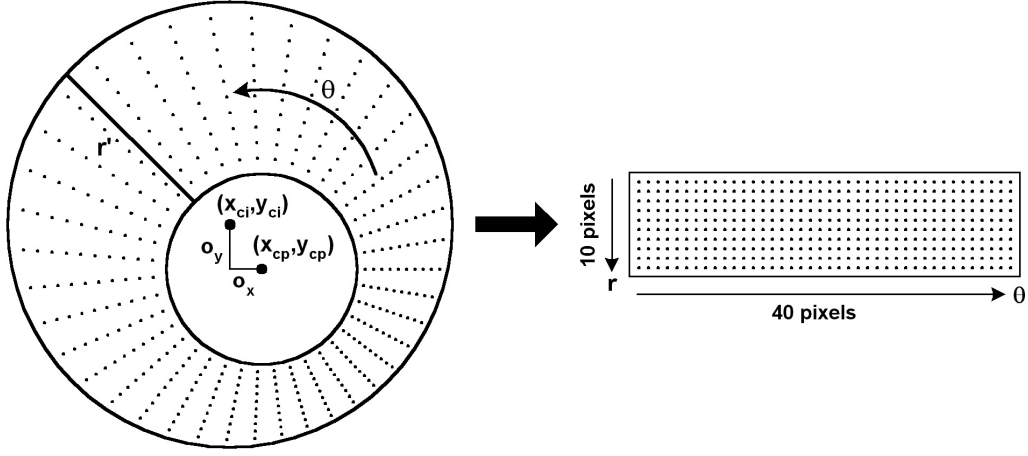


Figura 3.5: Esboço do processo de normalização com resolução radial igual a 10 pixels e resolução angular igual a 40 pixels.

$$r' = \sqrt{ab} \pm \sqrt{ab^2 - a - r_i^2} \quad (3.6)$$

sendo

$$a = o_x^2 + o_y^2$$

$$b = \cos \left(\pi - \arctan \left(\frac{o_y}{o_x} \right) - \theta \right)$$

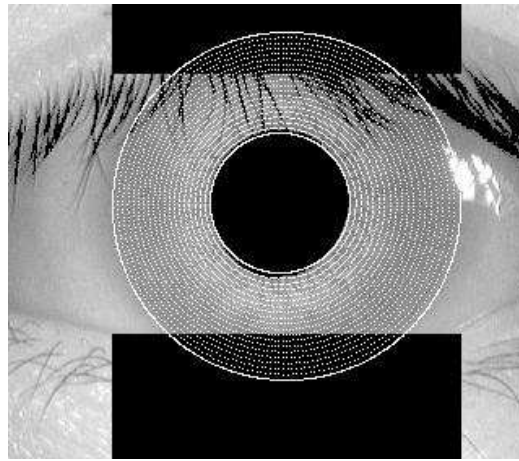
onde o_x e o_y representam o deslocamento do centro da pupila em relação ao centro da íris, r' é a distância entre as bordas da pupila e da íris para um determinado valor de θ e r_i é o raio da íris. A Equação (3.6) fornece o raio da região da íris em função do ângulo θ .

Para garantir uma dimensão constante na representação retangular, um número constante de pontos é selecionado ao longo de cada linha radial, independente da largura da região entre as bordas da pupila e da íris para determinado ângulo.

O método utilizado para a localização da íris pode não apresentar uma

precisão perfeita, assim, para tentar evitar que áreas não pertencentes à íris sejam utilizadas na representação normalizada, são desconsiderados os pontos da borda da pupila e da íris.

Uma vez obtidas as posições radial e angular dos pontos selecionados, são encontradas as coordenadas cartesianas correspondentes, utilizando-se a Equação (3.3). A saída normalizada é gerada ao se extrair os valores de intensidade de tons de cinza da imagem original nas posições cartesianas obtidas.



(a)



(b)



(c)

Figura 3.6: *a)Pixels selecionados na imagem original para a geração da representação retangular da íris; b)Representação retangular da região da íris com dimensões 18 x 240 pixels; c)Máscara de ruídos: a parte branca representa as regiões que sofrem interferência das pálpebras e cílios (dimensões: 18 x 240 pixels).*

Esse procedimento, além de levar em consideração o fato da pupila e da

íris não serem concêntricas, também elimina os problemas da dilatação da pupila e do *zoom* da imagem, gerando uma saída representativa da região da íris com dimensões constantes. Porém, este método não compensa as inconsistências rotacionais da imagem do olho. Neste trabalho, seguindo o procedimento sugerido por John Daugman [15, 16, 17, 18], tais inconsistências são tratadas na fase de comparação, onde o *template* de uma íris é deslocado na direção de θ até os *templates* de duas íris ficarem alinhados.

Ainda nesta fase, é gerada uma máscara de ruídos com a mesma dimensão da representação retangular da íris para marcar os pixels pertencentes às pálpebras e cílios que foram detectados na etapa de localização e que interferem na região da íris. Para gerar esta máscara de ruídos deve-se encontrar os pontos que foram identificados pelo valor *NaN* (*not a number*) na etapa de localização e marcá-los com o valor 1, enquanto todos os outros pixels são representados pelo valor 0. Na representação normalizada da íris, os pixels referentes aos ruídos são substituídos pelo valor de intensidade de tom de cinza médio dos outros pixels.

A Figura 3.6 ilustra o resultado da aplicação do algoritmo de normalização em uma imagem.

3.3 Método de Extração e Codificação de Características

Para a extração de características da íris, John Daugman [15, 16, 17, 18] empregou filtros de Gabor que são, tradicionalmente, utilizados para se obter, simultaneamente, uma localização espacial e de frequência da informação de um determinado sinal. Um filtro de Gabor é construído modulando-se uma

senóide/cossenóide com uma Gaussiana. Uma senóide/cossenóide pode ser perfeitamente localizada na frequência, mas não no espaço. A modulação de uma senóide/cossenóide com uma Gaussiana fornece uma localização no espaço, porém, com uma perda da localização da frequência. A decomposição do sinal é realizada utilizando-se um par de filtros de Gabor em quadratura, com a parte real representada por uma cossenóide modulada por uma Gaussiana e a parte imaginária representada por uma senóide também modulada por uma Gaussiana. A parte real e a imaginária são conhecidas, respectivamente, como componente de simetria par e componente de simetria ímpar que estão ilustradas na Figura 3.7.

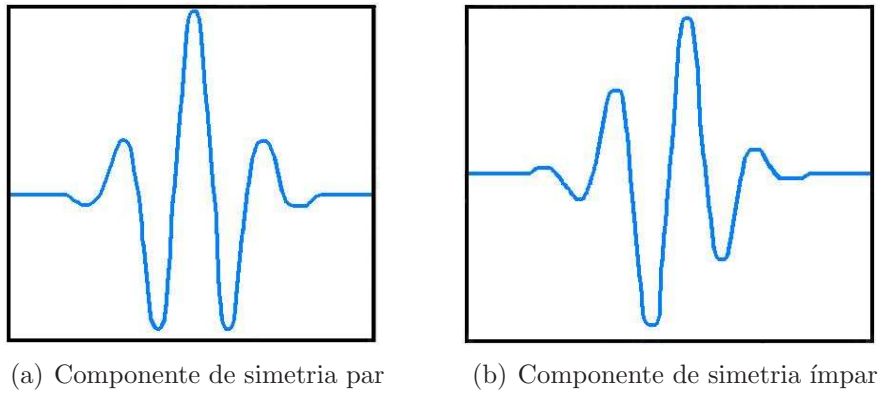


Figura 3.7: *Par de filtros de Gabor em quadratura.*

A frequência central é determinada pela frequência da senóide/ cossenóide e a largura de faixa do filtro é determinada pela largura da Gaussiana [50].

O filtro de Gabor, entretanto, possui algumas limitações. Sempre que a largura de faixa do filtro de Gabor for maior do que uma oitava, a componente de simetria par irá possuir uma componente dc [3]. Esta dificuldade pode ser vista observando-se a função de transferência da componente de simetria par no domínio da frequência (ver Figura 3.8). A função de transferência é a soma de duas Gaussianas centralizadas em mais e menos a frequência central. Se o desvio padrão dessas Gaussianas for maior do que aproximadamente um

terço da frequência central as caudas das duas Gaussianas começarão a se sobreporem na origem resultando em uma componente dc. Na situação limite onde a frequência central é igual a três vezes o desvio padrão, a largura de faixa será aproximadamente uma oitava [3].

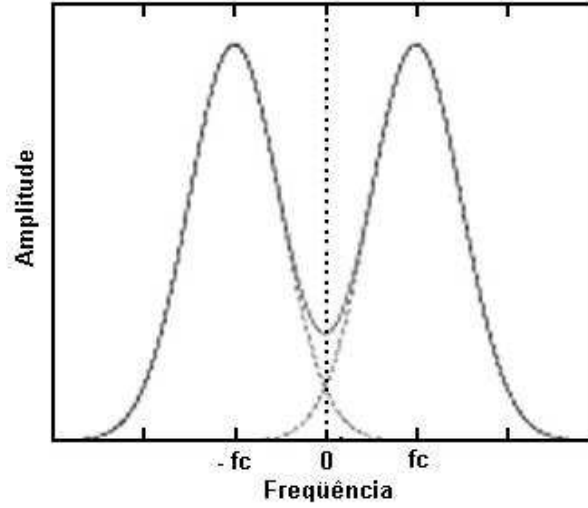


Figura 3.8: *Função de transferência da componente de simetria par do filtro de Gabor com uma largura de faixa grande. As duas gaussianas que formam a função se sobrepõem na origem resultando em uma componente dc significativa.*

Uma alternativa para resolver esse problema é o filtro Log-Gabor proposto por Field [30]. Neste caso, a Gaussiana é representada em uma escala logarítmica e assim, é possível obter uma componente dc igual a zero para qualquer que seja a largura de faixa utilizada. A resposta em frequência do filtro Log-Gabor é obtida pela Equação (3.7).

$$G(f) = \exp \left[\frac{-(\log(f/f_c))^2}{2(\log(\sigma/f_c))^2} \right] \quad (3.7)$$

onde f_c representa a frequência central e σ fornece a largura de faixa do filtro.

A extração das características foi realizada pela convolução da representação normalizada da íris com o filtro Log-Gabor 1D. Como a representação normalizada possui duas dimensões, ela deve ser dividida em sinais de uma

única dimensão para serem convoluídos com o filtro Log-Gabor 1D. Para isso, cada linha da representação normalizada 2D foi convoluída separadamente com o filtro. Cada linha corresponde a um anel circular na região da íris. A direção angular foi utilizada ao invés da direção radial (que corresponde às colunas da representação normalizada) porque uma maior independência das características ocorre na direção angular.

Os pixels das regiões de ruído detectados na etapa de localização são substituídos pelo valor de intensidade de tom de cinza médio dos outros pixels para evitar que eles interfiram na filtragem.

Para demodular a saída do filtro com o objetivo de codificar os dados foi empregado o método sugerido por Daugman [16]. A filtragem origina coeficientes complexos cujos valores das partes real e imaginária determinam as coordenadas de um fasor no plano complexo. A fase de cada fasor é quantizada para um dos quatro quadrantes do plano complexo como ilustrado na Figura 3.9. Foi mostrado por Oppenheim and Lim [56] que as informações de fase são melhores do que as de amplitude para fornecer as informações mais significativas de uma imagem, uma vez que na codificação são desconsideradas informações irrelevantes como a iluminação, que é representada pela amplitude.

Esses quatro níveis são representados utilizando-se dois bits de dados, assim, cada pixel da representação normalizada da íris corresponderá a dois bits de dados no *template* final da íris. A quantização de fase é realizada de forma que quando se passa de um quadrante para outro, somente um bit muda. Isto irá minimizar a quantidade de discordância entre os bits no momento da comparação no caso de duas íris provenientes da mesma pessoa estarem ligeiramente desalinhadas, resultando em um reconhecimento mais preciso.

Quando se utiliza apenas um filtro, a quantidade de bits de um *template*

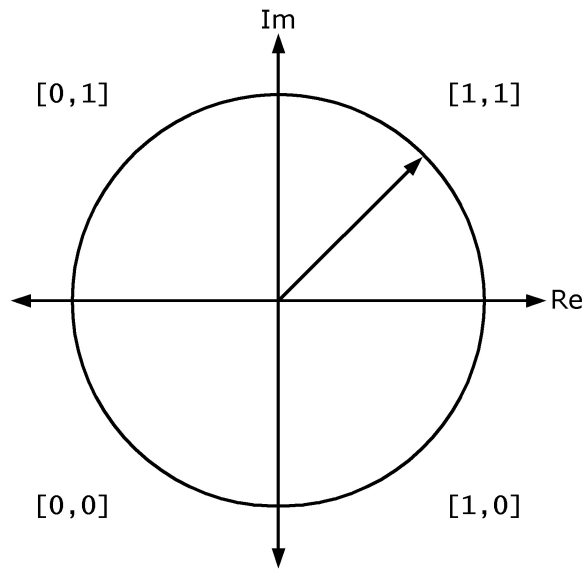


Figura 3.9: *Plano complexo utilizado para quantização.*

será a resolução angular vezes a resolução radial vezes dois bits. O processo de codificação gera, além do *template* contendo os bits de informações da íris, uma máscara de ruído correspondente que representa as áreas corrompidas. Essa máscara é gerada a partir daquela fornecida na fase de normalização e possui a mesma dimensão do *template*.

Para a implementação do processo é necessário definir a frequência central do filtro e os parâmetros da função Gaussiana. Estes valores devem ser atribuídos de forma a maximizar a eficiência do sistema de reconhecimento.

A Figura 3.10 mostra o *template* binário e a máscara de ruídos gerados a partir de uma imagem normalizada da íris.

3.4 Método de Comparação

O método de codificação utilizado gera *templates* binários, portanto, a utilização da distância de Hamming (dH) para processar a comparação

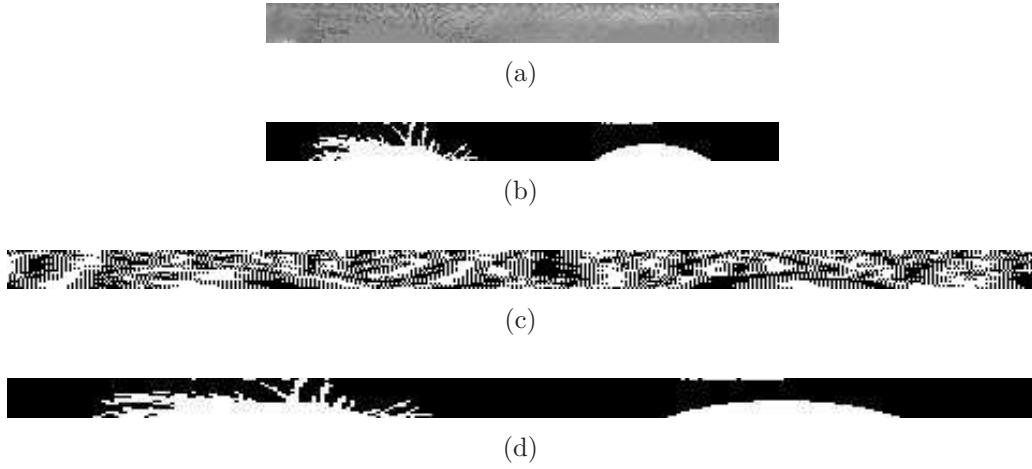


Figura 3.10: a) Representação normalizada da íris com dimensões 18 x 240 pixels; b) Máscara de ruídos com dimensões 18 x 240 pixels; c) Template binário com dimensões 18 x 480 pixels, gerado à partir da codificação de cada pixel da representação normalizada da íris, sendo que cada pixel é codificado por dois bits; d) Codificação da máscara de ruídos com dimensões 18 x 480 (cada pixel da máscara de ruídos é codificado por dois bits).

apresenta-se como uma alternativa interessante por ser própria para trabalhar com padrões binários. A distância de Hamming é uma medida quantitativa da variação entre os bits de dois *templates*. Esta medida é obtida pela comparação bit a bit dos *templates* seguida do cálculo da razão entre a quantidade de bits que não se correlacionam e a quantidade total de comparações entre os bits.

Desta forma, a dH pode ser definida pela Equação (3.8).

$$DH = \frac{1}{N} \sum_{j=1}^N \left(X_j \text{ (XOR) } Y_j \right) \quad (3.8)$$

onde a soma dos resultados da operação “ou-exclusivo” entre os bits correspondentes dos *templates* X e Y representa a quantidade de bits discordantes e N representa o total de bits do *template* utilizados para comparação.

Uma vez que a íris de um determinado indivíduo contém características

com um alto grau de liberdade, cada íris irá produzir um modelo binário independente do gerado por outra íris e por outro lado, dois códigos produzidos pela mesma íris serão fortemente correlacionados.

Na implementação foi empregado o algoritmo da distância de Hamming sugerido por John Daugman [15, 16, 17, 18]. Este algoritmo incorpora as informações de uma máscara de ruído gerada na fase de codificação, de forma que somente os bits significativos (realmente pertencentes à região da íris) sejam utilizados para o cálculo da distância de Hamming.

Assim, para a comparação entre dois *templates* só serão utilizados os bits que correspondem a um bit “0” nas máscaras de ruído das duas imagens a serem comparadas. Para implementar essa idéia e garantir que somente os bits significativos serão utilizados a equação da distância de Hamming deve ser modificada para a Equação (3.9).

$$DH = \frac{1}{N - \sum_{k=1}^N (MX_k(OR)MY_k)} \sum_{j=1}^N \left(X_j(XOR)Y_j(AND)\overline{MX_j}(AND)\overline{MY_j} \right) \quad (3.9)$$

onde X_j e Y_j são os bits dos *templates* a serem comparados, MX_j e MY_j são os bits das máscaras de ruído correspondentes a X_j e Y_j respectivamente, $\overline{MX_j}$ e $\overline{MY_j}$ são o complemento de MX_j e MY_j e N é o número total de bits de cada *template*.

Teoricamente, a equação da distância de Hamming pode gerar valores que vão de 0,0 (quando todos os bits dos *templates* comparados são concordantes) até 1,0 (quando todos os bits são discordantes). Assim, poderia ser esperado que a comparação de dois *templates* gerados a partir da mesma íris resultasse em uma dH igual a 0,0. Na prática, o valor obtido se aproxima de 0,0 mas

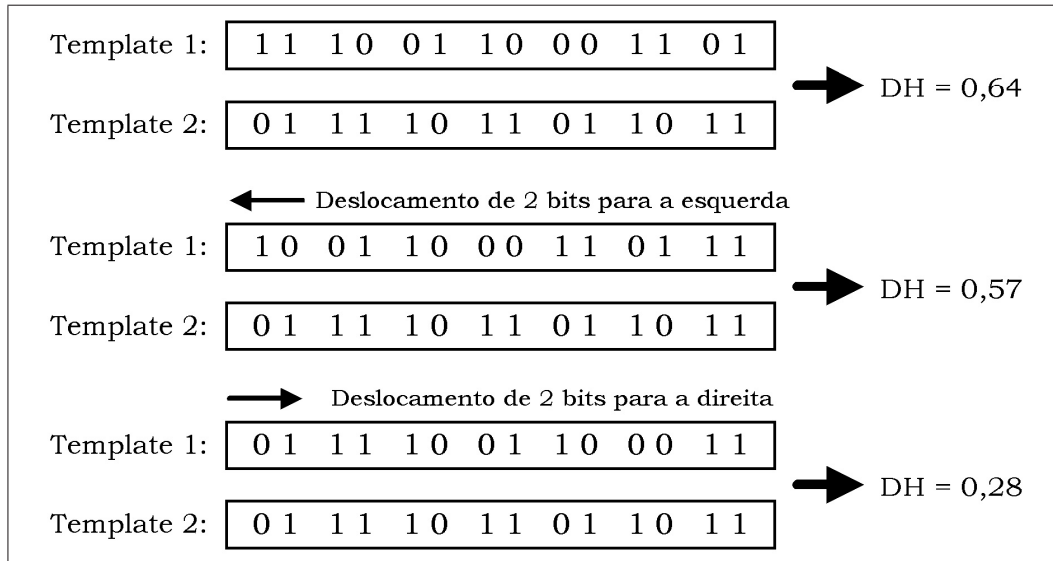


Figura 3.11: Representação de 1 deslocamento onde são deslocados sempre dois bits de cada vez, pois cada pixel da representação normalizada da íris é codificado por dois bits. Deve-se considerar somente a menor distância de Hamming que, neste caso, é igual a 0,28 e representa o melhor casamento entre os dois templates comparados.

não é exatamente igual a 0,0 uma vez que o processo de normalização não é perfeito, além de que ainda pode existir interferência de ruídos não detectados anteriormente. Estes fatores provocam algumas variações nos *templates* e estas variações estarão presentes no momento da comparação interferindo no cálculo da dH.

Quando se compara *templates* gerados a partir de íris diferentes, ou seja, modelos completamente independentes, obtém-se uma dH próxima de 0,5 e não de 1,0. Isso ocorre porque a independência implica que a correspondência entre dois bits será totalmente aleatória, ou seja, a probabilidade de um bit ser igual a 1 ou 0 é 50%. Portanto, teoricamente, a metade dos bits serão concordantes e a outra metade não, o que resulta em uma dH igual a 0,5.

Daugman [15, 16, 17, 18] sugeriu um método para corrigir o desalinhamento da representação normalizada de uma íris causado por variações rota-

cionais no momento da aquisição da imagem. Estas variações incluem uma possível rotação da câmera e/ou inclinação da cabeça. A minimização desses efeitos é processada na fase de comparação. Depois que a dH entre dois *templates* é calculada, um dos *templates* é deslocado para a esquerda ou para a direita. Este deslocamento na direção horizontal corresponde a uma rotação da região original da íris de um ângulo dado pela resolução angular usada na normalização. Durante cada deslocamento são movidos dois bits do *template*, uma vez que o processo de codificação gera dois bits de informação para cada pixel da região normalizada da íris. A cada deslocamento, os *templates* são comparados e um correspondente valor de dH é calculado. Ao final de sucessivos deslocamentos, é escolhido o menor valor de dH que corresponde ao melhor casamento entre os *templates*. O número de deslocamentos requeridos para lidar com as inconsistências rotacionais é determinado pela diferença angular máxima entre duas imagens da mesma íris. Um deslocamento é definido por um deslocamento para a esquerda e, em seguida, um deslocamento para a direita a partir da posição inicial. A Figura 3.11 ilustra o processo de um único deslocamento.

3.5 Bancos de Imagens de Íris

Neste trabalho foram utilizados dois bancos de imagens de íris. Algumas características dessas imagens são apresentadas a seguir.

3.5.1 Banco de imagens UBIRIS

O banco de imagens UBIRIS - Sessão 1 [63] contém um total de 1201 imagens de íris provenientes de 241 pessoas diferentes sendo, em torno de 5 imagens por pessoa. Estas imagens foram capturadas em um ambiente es-

curo com a intenção de minimizar os fatores de ruído, especialmente aqueles relativos a reflexos, luminosidade e contraste. Mesmo assim, todas as imagens apresentam reflexo especular próximo ou no interior da pupila e ainda, algumas delas estão desfocadas. Em geral, existe um bom contraste entre a esclera e a borda externa da íris, porém, o contraste entre a pupila e a região da íris é pequeno em muitas imagens, o que dificulta o processo de localização da íris.

As imagens foram convertidas para uma escala de cinza de 24 bits, possuem dimensão 200 x 150 (largura x altura) e foram armazenadas em formato JPEG com compressão sem perda. A Figura 3.12 mostra alguns exemplos de imagens do banco de imagens UBIRIS.

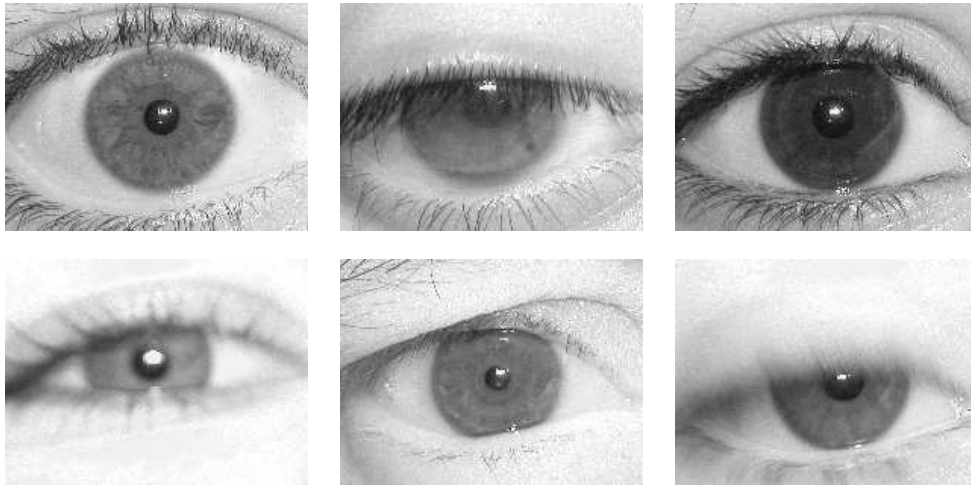


Figura 3.12: *Exemplos de imagens de íris do banco de imagens UBIRIS.*

3.5.2 Banco de imagens MMU

O banco de imagens MMU [2] contém um total de 450 imagens de íris provenientes de 45 pessoas diferentes. Foram capturadas 10 imagens de cada indivíduo, sendo 5 imagens diferentes de cada um dos olhos. Sabe-se que as

características do olho esquerdo de uma pessoa são diferentes das características do olho direito, portanto, quando se comparam duas imagens sendo cada uma proveniente de um dos olhos, mesmo sendo de uma mesma pessoa, espera-se que o sistema considere como sendo de pessoas diferentes. Assim, este tipo de comparação foi utilizada como comparação inter-classe.

As imagens do banco MMU foram capturadas utilizando a câmera semi-automática LG IrisAccess®2200, com luz visível e com uma distância do indivíduo na faixa de 7 a 25 cm. As imagens possuem pontos de reflexos especulares na pupila e, em geral, apresentam um bom contraste entre as regiões da parte branca do olho, da íris e da pupila.

As imagens foram armazenadas em escala de cinza, com dimensão 320 x 240 (largura x altura) e em formato Bitmap. A Figura 3.13 mostra alguns exemplos de imagens do banco de imagens MMU.

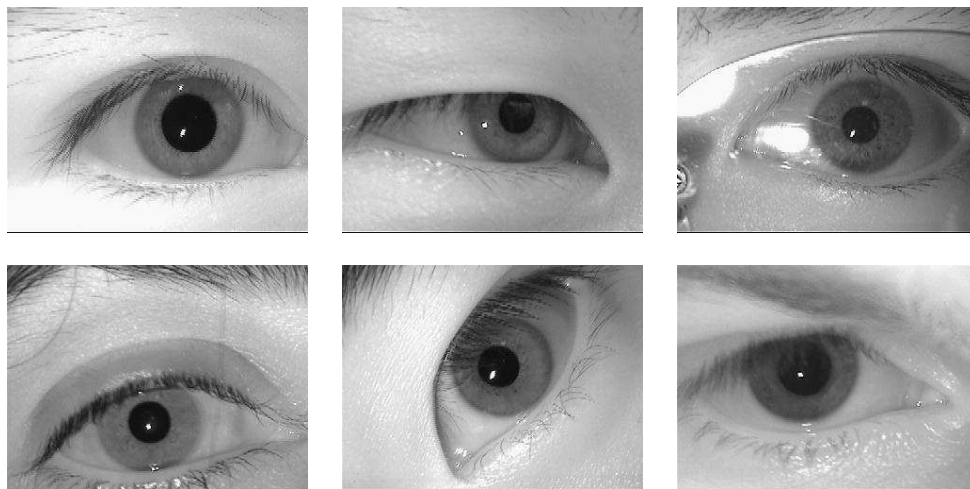


Figura 3.13: *Exemplos de imagens de íris do banco de imagens MMU.*

3.6 Resultados Experimentais

As implementações do sistema empregando técnicas tradicionais foram testadas utilizando-se dois bancos de imagens: UBIRIS [63] e MMU [2].

Existem diversos parâmetros variáveis no sistema, que são: resoluções radial e angular da representação retangular da íris, parâmetros do filtro Log-Gabor (frequência central do filtro e os parâmetros da função Gaussiana), quantidade de deslocamentos do *template* realizados para comparação e número de comparações realizadas para teste.

As resoluções da representação retangular e os parâmetros do filtro foram variados com o objetivo de encontrar o conjunto de parâmetros que maximiza a eficiência do sistema. Os demais parâmetros foram definidos com valores fixos.

Foram definidas duas variáveis para representar os parâmetros do filtro:

- minWaveLength: Comprimento de onda central que fornece a frequência central.
- sigmaOnf (σ/f): Largura de faixa do filtro.

Para processar a etapa de localização da região da íris, foi empregada a Transformada Circular de Hough para o banco de imagens UBIRIS e o Operador Integro-Diferencial para o banco de imagens MMU.

A seguir são apresentados os resultados obtidos, para cada banco de imagens, quando se variam os parâmetros.

3.6.1 Ajuste dos parâmetros para o banco de imagens

UBIRIS

Para o ajuste da resolução da representação retangular da íris, quando foi utilizado o banco de imagens UBIRIS, a eficiência do sistema foi medida para diversos valores de resolução angular e de resolução radial. A resolução angular foi variada de 40 até 180 pixels enquanto a resolução radial foi variada de 6 até 26 pixels. A porcentagem de erro de reconhecimento obtida para cada situação é mostrada na Tabela 3.1. Para efetuar essas simulações os demais parâmetros foram fixados com os seguintes valores:

Número de comparações intra-classe = 1758

Número de comparações inter-classe = 16290

Número de deslocamentos = 4 para a direita + 4 para a esquerda

sigmaOnf = 0,4

minWaveLength = 13

Também foram realizadas simulações variando os parâmetros do filtro Log-Gabor. A largura de faixa do filtro (sigmaOnf) foi variada de 0,2 até 0,9 enquanto o comprimento de onda central (minWaveLength) foi variado de 3 até 25. A Tabela 3.2 mostra a porcentagem de erro obtida para cada caso. Para efetuar estas simulações os demais parâmetros foram fixados com os seguintes valores:

Número de comparações intra-classe = 1758

Número de comparações inter-classe = 16290

Número de deslocamentos = 4 para a direita + 4 para a esquerda

Resolução Radial = 14

Resolução Angular = 150

Pode-se observar nas Tabelas [3.1](#) e [3.2](#) que o menor erro foi obtido quando a largura de faixa do filtro foi igual 0,4, o comprimento de onda central foi igual a 13 e a resolução da representação retangular da íris assumiu um dos seguintes valores: 14 x 150, 14 x 180, 18 x 160, 20 x 150, 20 x 170, 22 x 160 ou 22 x 170. Para garantir uma codificação mais compacta é melhor utilizar a resolução 14 x 150.

O menor erro obtido foi 0,07% que equivale a uma taxa de acerto de 99,93%.

Tabela 3.1: Ajuste da resolução da representação retangular da íris para o banco de imagens UBIRIS. Os valores da segunda linha representam a resolução radial em número de pixels e os valores da segunda coluna representam a resolução angular em número de pixels. Os demais valores da tabela são as taxas de erro obtidas.

		Resolução Radial										
		6	8	10	12	14	16	18	20	22	24	26
Resolução Angular	40	11,12%	10,28%	9,53%	8,54%	8,64%	8,39%	8,30%	8,35%	8,30%	7,96%	7,97%
	50	7,27 %	5,30%	5,62%	5,19%	4,56%	5,00%	4,49%	4,86%	4,30%	4,35%	4,24%
	60	4,07 %	3,15%	3,24%	3,00%	2,81%	2,61%	2,50%	2,79%	2,52%	2,41%	2,49%
	70	2,60 %	2,17%	2,02%	1,78%	1,91%	1,64%	1,72%	1,76%	1,58%	1,73%	1,63%
	80	1,63 %	1,21%	1,53%	1,17%	1,15%	0,97%	1,15%	1,14%	0,94%	1,06%	1,03%
	90	0,97 %	0,82%	1,03%	0,75%	0,76%	0,70%	0,80%	0,57%	0,75%	0,67%	0,68%
	100	0,77 %	0,74%	0,63%	0,58%	0,39%	0,49%	0,49%	0,37%	0,49%	0,35%	0,38%
	110	0,72 %	0,45%	0,48%	0,44%	0,31%	0,33%	0,36%	0,29%	0,28%	0,26%	0,27%
	120	0,48 %	0,29%	0,38%	0,26%	0,23%	0,32%	0,32%	0,19%	0,17%	0,29%	0,24%
	130	0,32 %	0,18%	0,29%	0,16%	0,11%	0,21%	0,23%	0,14%	0,14%	0,18%	0,16%
	140	0,24 %	0,25%	0,26%	0,16%	0,13%	0,19%	0,20%	0,10%	0,16%	0,21%	0,12%
	150	0,22 %	0,20%	0,14%	0,12%	0,07%	0,11%	0,15%	0,07%	0,10%	0,09%	0,08%
	160	0,18 %	0,28%	0,15%	0,09%	0,08%	0,10%	0,07%	0,08%	0,07%	0,11%	0,08%
	170	0,11 %	0,17%	0,16%	0,13%	0,11%	0,12%	0,12%	0,07%	0,07%	0,10%	0,08%
	180	0,18 %	0,21%	0,16%	0,13%	0,07%	0,12%	0,13%	0,06%	0,08%	0,08%	0,12%

Tabela 3.2: *Ajuste dos parâmetros do filtro Log-Gabor para o banco de imagens UBIRIS. Os valores da segunda linha representam a largura de banda do filtro (sigmaOnf) e os valores da segunda coluna representam o comprimento de onda central (minWaveLength). Os demais valores da tabela são as taxas de erro obtidas.*

		sigmaOnf							
		0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
minWaveLength	3	0,27%	0,89%	2,88%	8,15%	17,48%	30,21%	40,15%	49,45%
	4	0,27%	0,39%	1,33%	3,21%	5,87%	9,72%	16,59%	29,41%
	5	0,24%	0,28%	0,72%	1,90%	3,06%	4,68%	7,36 %	16,41%
	6	0,30%	0,18%	0,47%	1,06%	2,04%	3,07%	5,63 %	12,82%
	7	0,32%	0,18%	0,29%	0,55%	1,17%	2,37%	4,03 %	12,26%
	8	0,39%	0,18%	0,21%	0,40%	0,71%	1,66%	4,03 %	10,11%
	9	0,47%	0,18%	0,18%	0,20%	0,57%	1,27%	3,26 %	11,20%
	10	0,64%	0,20%	0,18%	0,17%	0,38%	1,05%	2,76 %	11,38%
	11	0,86%	0,19%	0,12%	0,17%	0,33%	0,84%	2,80 %	11,97%
	12	0,90%	0,23%	0,12%	0,18%	0,29%	0,97%	2,68 %	11,95%
	13	1,03%	0,26%	0,07%	0,08%	0,33%	0,92%	3,04 %	11,97%
	14	1,11%	0,35%	0,10%	0,11%	0,34%	1,04%	3,11 %	12,97%
	15	1,24%	0,43%	0,14%	0,20%	0,41%	0,91%	3,16 %	15,30%
	16	1,39%	0,59%	0,22%	0,24%	0,44%	1,15%	3,89 %	15,55%
	17	1,60%	0,70%	0,23%	0,32%	0,57%	1,52%	4,61 %	15,56%
	18	1,76%	0,82%	0,30%	0,35%	0,76%	1,82%	4,91 %	17,35%
	19	1,93%	0,83%	0,33%	0,45%	0,88%	2,27%	5,30 %	17,84%
	20	2,01%	0,96%	0,45%	0,51%	0,93%	2,46%	5,67 %	17,37%
	21	2,19%	0,99%	0,59%	0,58%	1,09%	2,64%	6,29 %	19,46%
	22	2,50%	1,08%	0,82%	0,62%	1,23%	2,97%	6,61 %	19,16%
	23	2,64%	1,19%	0,97%	0,78%	1,46%	3,14%	7,50 %	17,88%
	24	2,83%	1,31%	1,02%	0,88%	1,60%	3,39%	7,98 %	23,30%
	25	3,09%	1,49%	1,24%	1,06%	1,81%	3,52%	8,00 %	26,65%

3.6.2 Ajuste dos parâmetros para o banco de imagens

MMU

Um procedimento equivalente foi utilizado para fazer o ajuste dos parâmetros quando foi utilizado o banco de imagens MMU. A resolução angular foi variada de 50 até 450 pixels, enquanto a resolução radial foi variada de

6 até 30 pixels. A porcentagem de erro de reconhecimento obtida para cada situação é mostrada na Tabela 3.3. Para efetuar essas simulações os demais parâmetros foram fixados com os seguintes valores:

Número de comparações intra-classe = 632

Número de comparações inter-classe = 3321

Número de deslocamentos = 8 para a direita + 8 para a esquerda

sigmaOnf = 0,4

minWaveLength = 17

Para as simulações variando os parâmetros do filtro Log-Gabor, a largura de faixa do filtro (sigmaOnf) foi variada de 0,2 até 0,9 enquanto o comprimento de onda central (minWaveLength) foi variado de 3 até 25. A Tabela 3.4 mostra a porcentagem de erro obtida para cada caso. Para efetuar essas simulações os demais parâmetros foram fixados com os seguintes valores:

Número de comparações intra-classe = 632

Número de comparações inter-classe = 3321

Número de deslocamentos = 8 para a direita + 8 para a esquerda

Resolução Radial = 8

Resolução Angular = 370

Pode-se observar nas Tabelas 3.3 e 3.4 que o menor erro foi obtido quando a largura de faixa do filtro foi igual 0,4, o comprimento de onda central foi igual a 17, a resolução radial foi 8 pixels e a resolução angular foi 370 pixels.

O menor erro obtido foi 0,91% que equivale a uma taxa de acerto de 99,09%.

Tabela 3.3: Ajuste da resolução da representação retangular da íris para o banco de imagens MMU. Os valores da primeira linha representam a resolução radial em pixels e os valores da primeira coluna representam a resolução angular em pixels. Os demais valores da tabela são as taxas de erro obtidas.

		Resolução Radial														Resolução Angular		
		6	8	10	12	14	16	18	20	22	24	26	28	30				
50		17,53%	16,19%	16,40%	15,47%	16,28%	15,49%	15,24%	15,15%	15,63%	15,44%	14,94%	15,40%	15,70%				
70		12,68%	10,97%	10,88%	11,29%	10,89%	10,52%	10,25%	10,90%	10,44%	10,52%	10,13%	10,71%	10,75%				
90		8,53%	7,56%	7,02%	7,15%	7,19%	6,97%	7,09%	7,06%	7,06%	6,94%	7,29%	7,04%	7,07%				
110		6,41%	5,85%	6,21%	5,63%	5,81%	5,66%	5,66%	5,20%	5,41%	5,87%	5,67%	5,72%	5,66%				
130		4,96%	4,44%	4,42%	4,29%	4,30%	4,22%	4,12%	4,23%	3,94%	4,30%	4,06%	4,06%	4,06%				
150		3,16%	2,84%	2,72%	2,80%	3,00%	3,06%	2,99%	2,92%	2,92%	2,80%	2,53%	2,81%	2,72%				
170		2,54%	2,27%	2,15%	2,09%	1,97%	2,04%	2,14%	1,97%	2,07%	1,69%	1,94%	1,79%	1,93%				
190		1,89%	1,63%	1,65%	1,78%	1,62%	1,39%	1,45%	1,54%	1,75%	1,36%	1,33%	1,36%	1,42%				
210		1,85%	1,48%	1,68%	1,30%	1,56%	1,33%	1,21%	1,21%	1,36%	1,33%	1,33%	1,30%	1,27%				
230		1,70%	1,35%	1,76%	1,43%	1,73%	1,64%	1,61%	1,46%	1,60%	1,46%	1,55%	1,58%	1,55%				
250		1,66%	1,32%	1,57%	1,41%	1,44%	1,54%	1,42%	1,48%	1,30%	1,45%	1,39%	1,51%	1,51%				
270		1,70%	1,27%	1,36%	1,33%	1,73%	1,55%	1,30%	1,39%	1,27%	1,28%	1,46%	1,42%	1,36%				
290		1,49%	1,20%	1,47%	1,28%	1,61%	1,40%	1,35%	1,21%	1,29%	1,31%	1,35%	1,44%	1,35%				
310		1,71%	1,15%	1,38%	1,25%	1,41%	1,43%	1,12%	1,29%	1,18%	1,28%	1,27%	1,35%	1,28%				
330		1,82%	1,07%	1,41%	1,19%	1,51%	1,25%	1,15%	1,32%	1,25%	1,12%	1,33%	1,28%	1,19%				
350		1,76%	0,94%	1,36%	1,28%	1,45%	1,31%	1,06%	1,21%	1,18%	1,24%	1,28%	1,34%	1,18%				
370		1,74%	0,91%	1,39%	1,29%	1,45%	1,22%	1,03%	1,15%	1,22%	1,25%	1,12%	1,28%	1,18%				
390		1,89%	1,24%	1,50%	1,40%	1,56%	1,05%	0,92%	0,96%	1,09%	1,19%	1,03%	1,18%	1,12%				
410		1,82%	1,23%	1,70%	1,60%	1,73%	1,51%	1,29%	1,36%	1,35%	1,51%	1,38%	1,59%	1,51%				
430		1,93%	1,16%	1,60%	1,60%	1,71%	1,27%	1,13%	1,28%	1,37%	1,41%	1,37%	1,59%	1,40%				
450		1,94%	1,32%	1,73%	1,79%	1,55%	1,46%	1,31%	1,37%	1,25%	1,42%	1,31%	1,52%	1,27%				

Tabela 3.4: Ajuste dos parâmetros do filtro Log-Gabor para o banco de imagens MMU. Os valores da segunda linha representam a largura de banda do filtro (*sigmaOnf*) e os valores da segunda coluna representam o comprimento de onda central (*minWaveLength*). Os demais valores da tabela são as taxas de erro obtidas.

		sigmaOnf							
		0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
minWaveLength	3	1,79%	5,28%	38,07%	80,44%	97,69%	99,01%	97,74%	98,82%
	4	1,57%	2,56%	12,06%	50,75%	82,78%	94,01%	96,22%	96,80%
	5	1,72%	2,33%	5,34 %	22,56%	52,32%	73,94%	85,75%	90,50%
	6	1,88%	1,72%	3,05 %	9,13 %	26,27%	44,38%	58,49%	71,27%
	7	2,04%	1,76%	2,86 %	4,82 %	11,81%	23,85%	39,01%	54,06%
	8	2,04%	1,44%	2,52 %	3,77 %	7,31 %	13,80%	24,29%	42,27%
	9	2,08%	1,59%	2,08 %	3,40 %	5,60 %	9,76 %	16,81%	35,59%
	10	2,14%	1,51%	1,69 %	2,83 %	4,60 %	7,25 %	13,63%	27,98%
	11	2,27%	1,63%	1,47 %	2,24 %	4,29 %	6,68 %	10,73%	27,26%
	12	2,20%	1,54%	1,40 %	2,14 %	3,14 %	4,69 %	8,64 %	24,09%
	13	2,27%	1,33%	1,33 %	1,76 %	2,76 %	4,01 %	7,34 %	22,34%
	14	2,20%	1,36%	1,03 %	1,60 %	2,24 %	4,13 %	6,24 %	20,81%
	15	2,27%	1,30%	1,00 %	1,46 %	2,12 %	3,80 %	5,85 %	20,60%
	16	2,35%	1,48%	0,97 %	1,19 %	1,88 %	3,37 %	6,39 %	20,00%
	17	2,56%	1,47%	0,91%	1,10 %	1,42 %	2,73 %	6,41 %	19,28%
	18	2,71%	1,47%	0,96 %	1,12 %	1,62 %	2,75 %	5,72 %	19,06%
	19	2,62%	1,50%	0,97 %	1,21 %	1,39 %	2,63 %	5,79 %	18,89%
	20	2,62%	1,46%	0,96 %	1,32 %	1,72 %	3,10 %	6,12 %	19,41%
	21	2,81%	1,56%	1,12 %	1,34 %	1,83 %	3,31 %	6,75 %	20,27%
	22	3,15%	1,59%	1,19 %	1,29 %	1,90 %	3,48 %	7,61 %	22,84%
	23	3,35%	1,68%	1,26 %	1,31 %	1,97 %	3,65 %	7,61 %	24,45%
	24	3,55%	1,71%	1,32 %	1,37 %	2,26 %	3,78 %	7,45 %	23,09%
	25	3,46%	1,89%	1,38 %	1,41 %	1,95 %	4,04 %	8,01 %	21,86%

3.7 Conclusões do Capítulo

Neste capítulo foram descritas as técnicas implementadas para executar cada etapa de processamento do sistema de reconhecimento de íris. Em todas as etapas foram utilizadas técnicas tradicionais, que são comumente citadas e empregadas na literatura.

Para localizar a região da íris, foram utilizadas implementações de duas técnicas: Transformada Circular de Hough e o operador integro-diferencial proposto por Daugman. Na etapa de normalização foi utilizado o método do anel de borracha que faz uma amostragem uniforme de pixels na região da íris levando em consideração a não concentricidade do contorno circular da pupila e da borda externa da íris e representa a região da íris por um retângulo. Para extrair as características da estrutura da íris foi empregado um filtro Log-Gabor e os coeficientes complexos originados foram quantizados para um dos quatro quadrantes do plano complexo gerando um *template* binário. Para realizar as comparações de *templates*, foi utilizada como métrica a distância de Hamming.

O sistema tradicional implementado foi simulado utilizando dois bancos de imagens de íris. Na seção 3.5 as principais características destes bancos de imagens são apresentadas. Para cada banco de imagens, os parâmetros do filtro Log-Gabor e, ainda, a resolução da representação retangular da íris foram ajustados de forma a otimizar a eficiência do sistema de reconhecimento. Na seção 3.6 observou-se que para o banco de imagens UBIRIS a maior taxa de acerto obtida foi de 99,93% e para o banco de imagens MMU foi de 99,09%.

Capítulo 4

MÉTODO PROPOSTO PARA LOCALIZAÇÃO DA ÍRIS

Em geral, os métodos de localização da íris têm como objetivo encontrar os parâmetros que definem os círculos da pupila e da íris, que nem sempre são concêntricos. Encontrados esses parâmetros, a região da íris é determinada. Tendo essa idéia em mente, na sequência, os fundamentos da proposta são descritos e a utilização de algoritmos evolutivos é justificada.

Assim como a transformada de Hough utilizada por Wildes [78], o método proposto exige que a imagem original seja submetida a um algoritmo de detecção de bordas.

Existem várias técnicas de detecção de bordas na literatura e, assim como na abordagem de Wildes, neste trabalho foi utilizado o método de Canny [8]. É necessário obter uma imagem de bordas binarizada como a mostrada na Figura 4.1. Nesta figura, os pixels brancos representam os pixels de borda.

As características do mapa de borda dependem de parâmetros de limi-

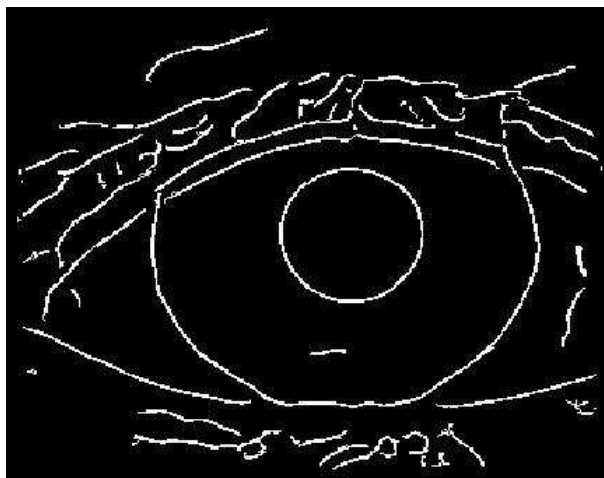


Figura 4.1: *Resultado da aplicação do algoritmo de detecção de bordas de Canny.*

arização do algoritmo de detecção de bordas. A variação desses parâmetros permite a detecção de um contraste maior ou menor na imagem. Por isso devem ser ajustados de forma a garantir que as informações de bordas mais importantes sejam adequadamente realçadas.

Este trabalho não tem o propósito de detalhar o processo de detecção de bordas nem investigar a influência do método de detecção de bordas ou do ajuste de seus parâmetros. Os parâmetros foram manualmente ajustados de forma a gerar mapas de bordas que, visualmente, continham as informações de bordas mais relevantes, ou seja, continham apenas os detalhes suficientes para perceber os contornos da íris.

Como sugerido por Wildes [78], somente as bordas verticais foram detectadas para a localização da borda externa da íris, enquanto ambos os gradientes horizontal e vertical foram igualmente ponderados para a localização da pupila.

Na Figura 4.1, pode-se verificar que os contornos da pupila e da íris são aparentes. Suponha que, utilizando a figura geométrica de um círculo (círculo imaginário), deseja-se sobrepor, na imagem de bordas, a maior quantidade de pixels brancos (pixels de borda) possível. É intuitivo que, para tanto, o

círculo imaginário deve ser posto sobre a pupila ou sobre a borda externa da íris.

A partir desse princípio, é possível encontrar os dois círculos desejados fazendo uma varredura na imagem de bordas. Essa varredura consiste em variar os parâmetros dos dois círculos imaginários dentro dos limites da imagem e, para cada conjunto de parâmetros $(x_c, y_c, radius)$, traçar o círculo correspondente sobre a imagem de bordas e contar a quantidade de pixels de borda que são sobrepostos. Assim, no final da varredura, o círculo imaginário que sobrepôs o maior número de pixels de borda é considerado o contorno da pupila ou da íris.

A busca por cada um dos círculos desejados deve ser realizada separadamente. A principal característica que diferencia o círculo da pupila e o círculo da íris é o tamanho do raio, já que as coordenadas do centro para esses dois círculos não são necessariamente as mesmas, mas estão sempre próximas.

As imagens do banco de dados utilizado (UBIRIS [63]) têm uma resolução de 200 pixels na direção horizontal (coordenada x) e 150 pixels na direção vertical (coordenada y). Para estas imagens, pode ser considerado que o raio da pupila varia entre 5 e 36 pixels, ao passo que o raio da íris varia de 40 a 71 pixels. Assim, para implementar uma varredura completa, a coordenada x_c deve variar de 1 a 200 e, para cada valor de x_c , a coordenada y_c deve variar de 1 a 150 e, ainda, para cada par de coordenadas do centro (x_c, y_c) , o raio deve variar de 5 a 36 para uma busca de pupila e de 40 a 71 para uma busca de íris.

Analisando esse procedimento, observa-se que, para encontrar a borda externa da íris, é necessário considerar $200 * 150 * (36 - 5) = 930.000$ círculos imaginários, e para encontrar a pupila, deve-se considerar outros $200 * 150 * (71 - 40) = 930.000$ círculos imaginários possíveis. Sob essas condições, esse procedimento é impraticável, pois exige um intenso processamento computa-

cional.

Esse tipo de problema, para o qual existe muitíssimas soluções e um vasto espaço de busca, é conhecido como Problema NP-completo. Existem algumas técnicas heurísticas capazes de abordar esses problemas. A proposta deste trabalho é empregar algoritmos meméticos (AM) para encontrar os parâmetros dos círculos da íris e da pupila com um tempo de processamento adequado, tornando o processo menos exigente computacionalmente. O Apêndice A apresenta uma introdução às definições de algoritmos genéticos (AG) e algoritmos meméticos (AM).

A aplicação dos AM para a localização da íris é descrita a seguir. Finalmente, são apresentados os resultados experimentais e uma comparação entre algoritmos.

4.1 Implementação do Método Proposto usando Algoritmos Meméticos.

Uma implementação de um algoritmo memético foi utilizada para resolver o problema mencionado.

Como explicado em detalhes anteriormente, o problema consiste em determinar as coordenadas do círculo que define a borda externa da íris e do círculo que define a borda da pupila.

Primeiramente, é necessário definir uma estrutura para os cromossomos que representarão possíveis soluções para o problema. Uma representação binária de 17 bits é utilizada, na qual a cada gene do cromossomo é atribuído um valor 0 ou 1, sendo que os primeiros 6 bits (ou genes) referenciam a coordenada x_c do centro do círculo, os próximos 6 bits referenciam a coordenada

y_c do centro e os últimos 5 bits representam o raio do círculo como ilustrado na Figura 4.2.

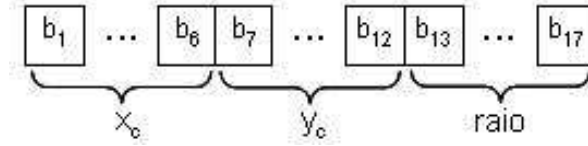


Figura 4.2: Estrutura do cromossomo utilizado pelo algoritmo memético.

A estrutura de 6 bits torna possível representar 64 valores diferentes de x_c e y_c . Como mencionado antes, o tamanho das imagens é 200 x 150, porém, em geral, a íris está posicionada na parte central da imagem do olho, não sendo necessário procurar por centros da íris ou da pupila nas extremidades da imagem. Então, a coordenada x_c é considerada como estando na faixa de 69 a 132 e a coordenada y_c é considerada na faixa de 33 a 107. O raio é representado por uma estrutura de 5 bits e, assim, 32 diferentes valores podem ser obtidos. Para a pupila, os 32 diferentes valores estão entre 5 e 36, e para a íris estão entre 40 e 71.

Um outro passo extremamente importante é a definição de uma função de aptidão que atribui uma pontuação (score) para cada indivíduo. Para o problema em questão, a aptidão de um indivíduo é obtida pelo número de pixels de borda que são sobrepostos pelos pixels do círculo representado pelo cromossomo. Assim, quanto maior o número de pixels sobrepostos, mais apto é o indivíduo.

O algoritmo deve gerar uma população inicial representando o espaço de busca inicial. A criação da população inicial é realizada atribuindo-se, aleatoriamente, um valor binário para cada gene de um cromossomo.

Duas populações de indivíduos foram criadas, uma constituída de candidatos a representar a borda externa da íris e outra para representar a borda da pupila. Essas populações são processadas independentemente.

Os indivíduos de uma população devem ser recombinados para criar uma nova geração. Isto é realizado selecionando-se pares de indivíduos. Um procedimento chamado de “Método da Roleta” [33] é utilizado. Este método baseia-se na teoria da sobrevivência dos mais aptos, isto é, indivíduos com uma aptidão maior têm maiores chances de serem selecionados para reprodução.

Dois operadores genéticos comuns foram usados: cruzamento e mutação. No primeiro caso, um par de indivíduos é selecionado da população e seus cromossomos são cortados em uma posição aleatória; isto gera duas caudas que são trocadas entre os dois cromossomos de forma que dois novos e diferentes indivíduos são gerados. A mutação pode ser aplicada a todo e qualquer descendente depois do cruzamento. Ela altera alguns genes de alguns cromossomos de forma aleatória e com uma probabilidade baixa.

Antes de proceder com a próxima geração, uma busca local é realizada para proporcionar a “evolução cultural”. A definição do mecanismo de busca local depende da representação da solução e de características intrínsecas ao problema. Quanto maior o conhecimento sobre o problema, mais eficientemente a busca local pode ser implementada. Na próxima seção, é descrito o mecanismo de busca local utilizado neste trabalho para implementar o algoritmo memético.

4.2 Mecanismo de Busca Local Implementado

A observação do desempenho da parte evolutiva do algoritmo (sem uso de busca local) demonstra que, muitas vezes, soluções permanecem presas a mínimos locais ao longo de várias gerações. Gráficamente, a situação é como a mostrada na Figura 4.3.

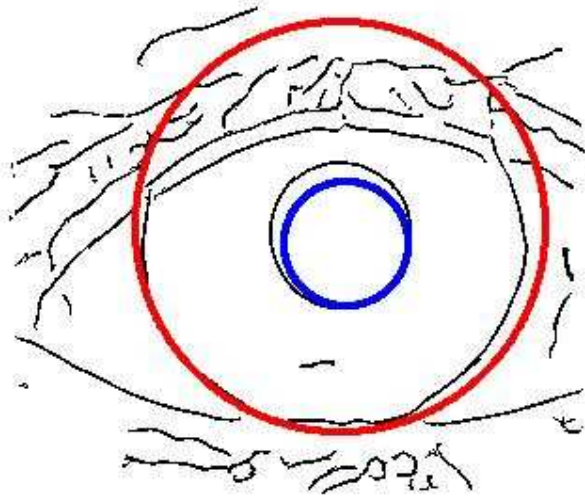


Figura 4.3: *Problema de mínimo local que frequentemente ocorre na execução do AG.*

Pode-se observar na Figura 4.3 que os círculos imaginários que representam a solução em questão estão muito próximos de uma solução ótima, mas o algoritmo evolutivo somente poderia gerar um indivíduo com aquelas características ótimas após muitas gerações. É nesse ponto que a busca local torna-se vantajosa e acelera a convergência do algoritmo.

A vizinhança da solução poderia ser investigada deslocando-se o centro do círculo para a direita e para a esquerda (coordenada x), para cima e para baixo (coordenada y) e, para cada uma dessas variações, o raio também pode ser modificado. Contudo, essa forma de busca local é computacionalmente onerosa, especialmente quando a população contém um grande número de indivíduos.

Por essa razão, para diminuir o esforço computacional da busca local, uma estratégia para reduzir a vizinhança foi proposta e implementada neste trabalho, de modo a guiar as coordenadas do círculo. Visualmente, nota-se que a solução da Figura 4.3 é tal que o círculo toca a borda da pupila em uma certa extensão, isto é, alguns pixels do círculo imaginário coincidem

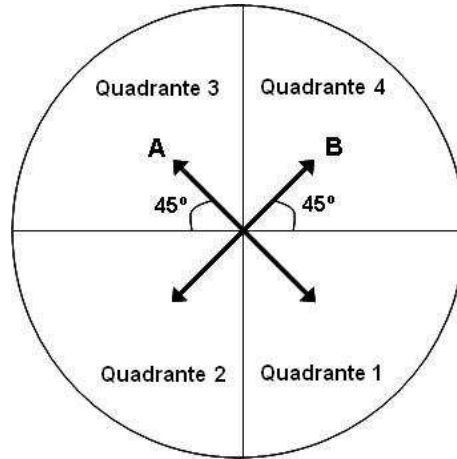


Figura 4.4: *Divisão do círculo em quadrantes e direção das linhas diagonais a serem selecionadas durante a busca local.*

com pixels da borda da pupila. Esta informação pode ser útil para guiar as coordenadas do círculo.

O mecanismo de busca local sugerido está estruturado no pseudo- código 4.2.1. A busca local é aplicada a todos os indivíduos da população (linha 3), lembrando que cada indivíduo armazena os parâmetros para definição de um círculo. Um círculo pode ser dividido em quadrantes e duas linhas diagonais são definidas conforme mostrado na Figura 4.4. Como já foi mencionado, para avaliar a aptidão de um indivíduo, soma-se a quantidade de pixels de bordas que coincidem com o círculo imaginário quando este é plotado sobre a imagem do olho. O algoritmo de busca local verifica em qual dos quadrantes ocorre a maior quantidade de pixels coincidentes (linha 6). Se um dos quadrantes, 1 ou 3 possuírem a maior quantidade de pixels coincidentes, o algoritmo irá variar as coordenadas do centro x_c e y_c mantendo o centro do círculo imaginário sempre sobre a diagonal A (linhas 8-9 e 12-13). Se, por outro lado, a maioria do pixels coincidentes estiverem nos quadrantes 2 ou 4, o centro do círculo imaginário será deslocado de forma a permanecer sobre a diagonal B (linhas 10-11 e 14-15).

```

1  // Procedimento BuscaLocal(NovaPop)
2
3  PARA cada indivíduo da população
4      REPETIR
5          Plotar na imagem de bordas, círculo representado pelo indivíduo;
6          Descobrir qual quadrante do círculo coincide com a maior ...
7          ... quantidade de pixels de borda;
8          SE a maior quantidade de pixels coincidentes está no quadrante 1
9              DiagonalSelecionada = diagonal A;
10         SE a maior quantidade de pixels coincidentes está no quadrante 2
11             DiagonalSelecionada = diagonal B;
12         SE a maior quantidade de pixels coincidentes está no quadrante 3
13             DiagonalSelecionada = diagonal A;
14         SE a maior quantidade de pixels coincidentes está no quadrante 4
15             DiagonalSelecionada = diagonal B;
16         i = 1;
17         ENQUANTO i <= VariaCentro FAÇA
18             Deslocar centro do círculo ao longo da DiagonalSelecionada;
19             j = 1;
20             ENQUANTO j <= VariaRaio FAÇA
21                 Variar o raio;
22                 NovaAptidao = Calcular a aptidão do novo indivíduo...
23                 ...que possui um novo centro e um novo raio.
24                 SE NovaAptidao > Aptidao do indivíduo original
25                     Indivíduo_Original = Novo_Indivíduo;
26                 j = j+1;
27             fimENQUANTO
28             i = i+1;
29         fimENQUANTO
30     ATÉ que nenhuma melhora seja obtida durante a iteração.
31 fimPARA

```

Código 4.2.1: *Procedimento da busca local.*

Portanto, a avaliação da vizinhança é realizada deslocando o centro do círculo imaginário um número de vezes igual a “*VariaCentro*” ao longo da diagonal selecionada (linhas 17 e 18), e, para cada posição assumida, o raio do círculo é variado um número de vezes igual a “*VariaRaio*” (linhas 20 e 21). Para cada situação, o novo valor de aptidão é calculado (linha 22). Se o novo valor de aptidão for maior do que o valor original, o indivíduo original é substituído pelo novo indivíduo (linhas 24 e 25). Todo esse processo se repete para o mesmo indivíduo até que não se encontre nenhum indivíduo melhor do

que o original durante uma execução completa (linha 30). A Figura 4.5 usa o círculo que representa a borda da pupila para ilustrar esse procedimento.

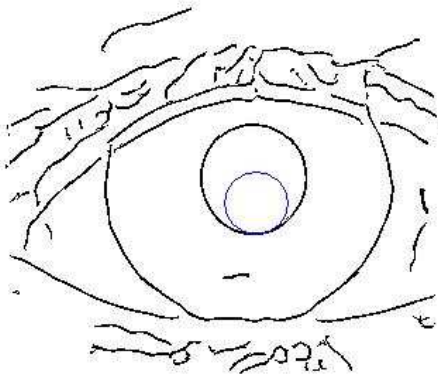
A Figura 4.5(a) mostra uma solução obtida pelo algoritmo de evolução genética. A Figura 4.5(b) ilustra o efeito do deslocamento do centro do círculo imaginário. Depois do deslocamento do centro do círculo, o raio é variado, como mostrado na Figura 4.5(c). Como foi possível encontrar uma solução melhor do que a solução original, o novo indivíduo substitui o indivíduo original e é executada uma nova iteração do algoritmo de busca local. Novamente, a quantidade de pixels coincidentes de cada quadrante é verificada, novas coordenadas para o centro do círculo são geradas (veja Figura 4.5(d)) e novos valores de raio são investigados (veja Figura 4.5(e)). O processo finaliza quando não ocorre melhora na aptidão do indivíduo.

Pode-se verificar que a busca local altera as características do indivíduo melhorando sua aptidão. Este indivíduo volta à população para ser submetido ao processo de evolução genética com uma aptidão melhor do que a que ele tinha anteriormente. Esta melhora ocorre devido à evolução cultural realizada pela busca local, que explora os conhecimentos disponíveis a respeito do problema.

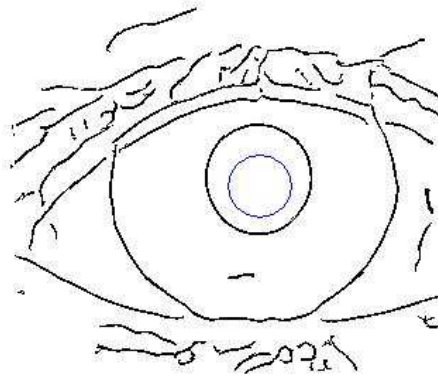
4.3 Definição dos Parâmetros

Como explicado no apêndice A, a implementação do AM requer a definição dos seguintes parâmetros: tamanho da população, número de gerações, taxa de cruzamento e taxa de mutação.

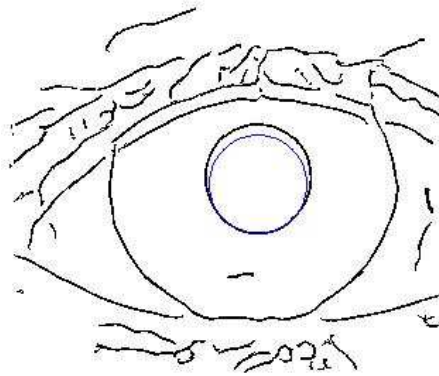
Para definir o conjunto de parâmetros mais adequado, o algoritmo foi aplicado inúmeras vezes a todas as imagens do banco de imagens e o seu comportamento foi analisado à medida que os parâmetros iam sendo variados. Durante as simulações, o tamanho da população foi variado uniformemente



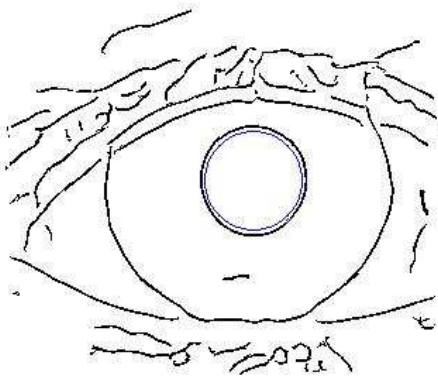
(a) Solução obtida pelo algoritmo de evolução genética.



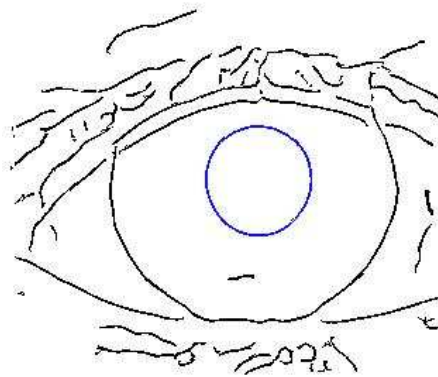
(b) Aplicando busca local: deslocando o centro do círculo.



(c) Aplicando busca local: variando o raio do círculo.



(d) Outra iteração da busca local: deslocando o centro.



(e) Variando o raio novamente.

Figura 4.5: *Ilustração do procedimento de busca local.*

de 10 até 200 indivíduos. Para um número maior de indivíduos, o tempo de processamento se torna inconvenientemente longo. As taxas de cruzamento e de mutação foram variadas uniformemente de 0% a 100%. O tempo de convergência do algoritmo foi a métrica utilizada para avaliar os parâmetros.

Os melhores resultados foram alcançados utilizando-se uma população de 50 indivíduos, uma taxa de cruzamento de 90% e uma taxa de mutação de 20%. Foi verificado que, para a maioria das imagens, a região da íris era corretamente segmentada até a quinta geração. Então, o número de gerações foi fixado em oito, para tentar garantir que um bom resultado seja alcançado. O número de gerações foi utilizado como critério de parada do AM.

4.4 Resultados Experimentais

Os testes experimentais do método proposto para localizar a região da íris foram realizados utilizando-se um total de 1201 imagens de olho do banco de imagens UBIRIS (Sessão 1) [63].

A seguir serão apresentados os resultados obtidos com o método proposto e será realizada uma comparação com o método de Wildes que utiliza a Transformada Circular de Hough que é o método de localização baseado em fronteira mais tradicional e mais empregado na literatura.

4.4.1 Avaliando o método proposto

Sistemas baseados em técnicas heurísticas, como é o caso dos Algoritmos Meméticos, são não determinísticos. Isto significa que uma solução diferente pode ser obtida cada vez que o sistema é simulado. Depois de definir os parâmetros variáveis do algoritmo (tamanho da população = 50 indivíduos; taxa de cruzamento = 90% e taxa de mutação = 20%), o sistema foi simulado inúmeras vezes. Em cada simulação, o algoritmo era aplicado a todas

as 1201 imagens e a quantidade de imagens que eram corretamente segmentadas era verificada. Depois que todas as simulações foram concluídas, foram calculados a média aritmética e o desvio padrão dos resultados obtidos.

Considerando todas as simulações que foram executadas, a quantidade média de imagens que foram segmentadas corretamente foi 1166 das 1201 imagens; somente 35 imagens não foram segmentadas corretamente. Isto significa que, quando o algoritmo é aplicado às imagens, em média, 97,08% das segmentações são aceitáveis.

O desvio padrão, que mede o espalhamento dos dados em torno da média, foi igual a 2,22. Este valor pode ser considerado pequeno, o que significa que, para várias simulações, a eficiência do algoritmo fica sempre próxima da eficiência média.

Os resultados mostraram que o método é robusto o suficiente para lidar com condições não ideais uma vez que altas taxas de acerto foram obtidas mesmo com imagens de baixo contraste e com a presença de reflexos especulares.

4.4.2 Comparação entre o método proposto e o método de Wildes

O método de segmentação de Wildes, que utiliza a tradicional Transformada de Hough Circular (THC), foi implementado e também foi aplicado às mesmas 1201 imagens do banco de imagens UBIRIS. Este algoritmo obteve sucesso em localizar a região da íris em 1119 imagens, o que significa uma eficiência de 93,17%. A comparação entre os resultados obtidos com o método proposto e o de Wildes é resumida na Tabela [4.1](#).

Outra vantagem do método proposto é o tempo de processamento. O principal problema da Transformada de Hough é que ela demanda um esforço computacional muito grande, enquanto que o método proposto, uma

Tabela 4.1: *Comparação dos resultados obtidos com o método proposto e o de Wildes.*

	Método Proposto	Método de Wildes
Segmentadas corretamente	1166 imagens	1119 imagens
Não segmentadas	35 imagens	82 imagens
Taxa de segmentação	97,08%	93,17%
Taxa de erro	2,92%	6,83%

vez que ele converge progressivamente a cada geração, exige muito menos processamento. Com isto, o tempo que o método proposto gasta para segmentar a íris é cerca de cinco vezes menor do que o tempo gasto pelo método de Wildes.

Uma característica comum aos dois métodos é a necessidade de ajuste de parâmetros para fazer a detecção de bordas. Algumas vezes isto pode ser considerado uma desvantagem pois, se as características de definição e contraste das imagens variarem, os parâmetros terão que ser modificados. Entretanto, deve-se levar em consideração o fato de que, para uma aplicação específica, as imagens são capturadas sob as mesmas condições e, por essa razão, os parâmetros precisam ser ajustados somente uma vez, garantindo uma boa eficiência do algoritmo.

4.5 Conclusões do Capítulo

Neste capítulo foi apresentado o método proposto para localizar a região da íris. A principal inovação desse método é a utilização de algoritmos evolutivos nessa etapa de processamento.

Inicialmente, a imagem é submetida a uma técnica de detecção de bordas de forma que os contornos circulares da pupila e da borda externa da íris fiquem aparentes. Então, um algoritmo memético (AM) é projetado para

encontrar os parâmetros de dois círculos que sobrepõem a maior quantidade de pixels de borda possível e que irão definir a região da íris.

Todos os detalhes da implementação do método e da definição de parâmetros são descritos. Na seção 4.2 é apresentado o mecanismo de busca local utilizado pelo AM.

Na seção 4.4 o método proposto é simulado, avaliado e comparado com o método de Wildes que utiliza a Transformada de Hough Circular. O novo método apresentou uma maior eficiência na localização da íris e ainda, exigiu um menor processamento computacional.

Capítulo 5

MÉTODO PROPOSTO PARA AMOSTRAGEM NÃO UNIFORME DE PIXELS NA REGIÃO DA ÍRIS

Como visto anteriormente neste trabalho, a imagem da íris é processada de forma que seja possível representar suas principais características por um *template* que pode ser binário ou numérico. Quando um indivíduo é cadastrado em um sistema de reconhecimento de íris para obter acesso a algum local ou informação, este *template* é gerado e armazenado computacionalmente ¹. Quanto maior a quantidade de informações codificadas, maior o espaço em disco necessário para armazenamento.

Tanto no processo de verificação quanto no processo de identificação, são realizadas comparações entre os *templates* gerados para fazer o reconhecimento do indivíduo. O tempo de processamento computacional dessas comparações é maior, quanto maior for o tamanho do *template*.

Para minimizar os problemas de espaço de armazenamento em disco e tempo de processamento das comparações, surgiu o desafio de se codificar a menor quantidade possível de características da íris, porém, sem comprometer

¹Atualmente, alguns sistemas armazenam a imagem do olho ao invés do *template*.

ter a confiabilidade do sistema.

Na literatura é possível identificar poucos trabalhos com esse objetivo. Estes poucos trabalhos buscam novas alternativas para melhorar a representação das características da íris aplicando-se novas técnicas para extração e codificação dessas características. Pode-se citar como exemplo o trabalho de Dey e Samanta [23] que aplica a transformada Wavelet Daubechies 4 para extrair as características da íris e representá-las em um *template* binário de tamanho reduzido. Ali et al. [6] fizeram a mesma coisa, porém, utilizando a transformada Wavelet Haar.

Neste trabalho levou-se em consideração o fato de que, antes que os algoritmos de extração e codificação das características da íris sejam aplicados, é necessário haver um processo de normalização que faz uma seleção (amostragem) de pixels na imagem e, geralmente, produz uma representação retangular da região da íris. Este processo de amostragem é de extrema importância, pois, é com ele que as características da íris ficam disponíveis para serem extraídas e codificadas.

Em geral, a amostragem de pixels para normalização é realizada de maneira uniforme ao longo de toda a região da íris. Este processo se torna parcialmente aleatório, pois, não se controla se os pixels selecionados armazenam mais ou menos informações.

Propõe-se, então, um método que permite ao sistema identificar e amostrar os pixels que armazenam uma maior quantidade de informações capazes de diferenciar um certo grupo indivíduos. O resultado dessa amostragem mais inteligente é que somente as informações mais relevantes são selecionadas, o que pode levar o sistema a dois benefícios principais:

- A geração de *templates* menores sem comprometer a precisão do reconhecimento. Isto é conseguido uma vez que, a codificação de uma

menor quantidade de características, por estarem melhor selecionadas, é suficiente para manter a capacidade de diferenciar os indivíduos; e

- O aumento da confiabilidade do sistema. Quando não se tem a preocupação de reduzir muito o tamanho do *template*, a amostragem dos “melhores” pixels aumenta a capacidade do sistema de diferenciar os indivíduos.

É possível encontrar diversos trabalhos que buscam novas alternativas para melhorar a representação das características da íris com o objetivo de aumentar a confiabilidade do sistema. Porém, na grande maioria das vezes, essas alternativas envolvem a aplicação de novas técnicas para extração e codificação dessas características, como por exemplo, a utilização de diferentes filtros ou de diferentes ferramentas de análise de textura, que é o caso dos trabalhos [39, 43, 46, 48, 65, 75, 78] dentre outros. Alguns outros trabalhos utilizam diferentes alternativas na tentativa de melhorar a eficiência do sistema. Pode-se citar como exemplo Dey e Samanta [23] que estabeleceram diferentes pesos para diferentes partes da região da íris no momento da comparação. Já Roy e Bhattacharya [65] propuseram a utilização de algoritmos genéticos para selecionar um melhor subconjunto de características da íris à partir das características extraídas por quatro diferentes algoritmos de extração de características.

Como mencionado acima, o método que é apresentado neste trabalho propõe selecionar os pixels localizados em posições específicas da região da íris de forma a permitir uma melhor representação das características que distinguem determinados indivíduos. Esta seleção de pixels é realizada utilizando-se algoritmos genéticos (AG) e origina uma representação retangular da região da íris que é, posteriormente, submetida aos algoritmos de extração e codificação de características. A descrição dos algoritmos genéticos é apre-

sentada no Apêndice A.

A grande novidade desse método é que ele é aplicado em uma etapa de pré-processamento, isto é, considerando um sistema de reconhecimento de íris que é utilizado na prática como instrumento de controle de acesso, o algoritmo proposto é executado antes de se colocar o sistema em funcionamento e depois que os usuários são cadastrados. Explicando, o algoritmo genético utiliza as imagens do banco de dados para fazer uma busca pela posição dos pixels que, quando processados pelos algoritmos de extração de características, codificação e comparação, tornarão o sistema mais eficiente. Uma vez que o AG obtém sua solução, esta informação é adicionada ao sistema de forma que, durante a sua execução, todas as imagens que forem processadas terão amostrados os pixels localizados nas posições escolhidas pelo AG.

Esse método pode ser utilizado como um pré-processamento de qualquer sistema cujo algoritmo envolve a representação da região da íris de forma retangular, que é o caso da grande maioria dos métodos encontrados na literatura. Isto significa que qualquer método de extração de características, seja utilizando algum filtro ou alguma transformada ou qualquer técnica de análise de textura, pode ser beneficiado por essa etapa de pré-processamento.

Nas próximas seções serão apresentados os detalhes do método proposto, assim como os resultados experimentais obtidos.

5.1 Método Proposto

O processo de geração da representação retangular da íris proposto por Daugman, que foi mostrado na seção 3.2), é o mais utilizado na literatura e sugere uma amostragem uniforme de pixels ao longo de toda a região anelar da íris. São traçadas linhas radiais uniformemente espaçadas à partir do centro da pupila e uma determinada quantidade de pixels uniformemente es-

paçados são coletados em cima dessas linhas. A quantidade de linhas traçadas define a resolução angular e o número de pixels amostrados em cada linha define a resolução radial que juntas definem a resolução da representação retangular da íris.

Em trabalhos desenvolvidos anteriormente [59, 60, 61], foram realizados experimentos para avaliar a influência de algumas partes da íris no reconhecimento de indivíduos. Ficou constatado que as regiões da íris mais próximas da pupila armazenam mais informações relevantes. Nesses trabalhos também foram realizados procedimentos, inclusive utilizando algoritmos genéticos, para verificar o efeito de se amostrar os pixels de forma não uniforme. Nestes procedimentos, a não uniformidade foi implementada apenas na direção radial e não se considerou a possibilidade de reduzir a resolução da representação retangular com o objetivo de produzir *templates* menores. Os bons resultados obtidos com relação a melhora no desempenho do sistema, serviram de motivação para o aprimoramento da idéia e a investigação de novas possibilidades.

O novo método sugere que, inicialmente, utilizando o procedimento tradicional de Daugman, seja gerada uma imagem normalizada (representação retangular da região da íris) com resoluções radial e angular bem maiores do que a que se deseja processar pelo algoritmo de extração de características. Um algoritmo genético devidamente projetado é então aplicado com o objetivo de escolher de quais linhas radiais e de que posições em cima dessas linhas serão amostrados os pixels que serão processados. O algoritmo genético deve ter a informação prévia da resolução da representação retangular desejada para determinar a quantidade de linhas radiais e a quantidade de posições em cima dessas linhas que poderão ser selecionadas. A solução do algoritmo genético origina uma representação retangular com a resolução desejada e contendo os pixels que, quando amostrados e processados, otimizam a eficiência do

sistema de reconhecimento.

A utilização de algoritmos genéticos para resolver o problema em questão é justificada pela enorme quantidade de possíveis soluções. Para se ter uma noção da dimensão de um possível espaço de busca será apresentado um exemplo prático. Supõe-se uma situação em que a resolução da imagem normalizada gerada inicialmente é 30 x 250 pixels e a resolução da representação retangular produzida pelo AG é 14 x 150. Neste caso, de um total de 250 linhas radiais o AG pode escolher qualquer combinação de 150 linhas, e, de um total de 30 posições em cima das linhas radiais o AG pode escolher qualquer combinação de 14 posições. Com isso, a quantidade de soluções possíveis pode ser calculada utilizando-se a Equação 5.1.

$$C_{150}^{250}x C_{14}^{30} = \binom{250}{150}x \binom{30}{14} = \frac{250!}{150!(250-150)!}x \frac{30!}{14!(30-14)!} \cong 8.8x10^{79} \quad (5.1)$$

Trata-se de um espaço de busca extremamente vasto, portanto, esse é um problema que tipicamente, os Algoritmos Genéticos estão preparados para resolver, combinando as informações e buscando por uma solução que, não necessariamente será a ótima, mas que será adequada ao problema.

Na próxima seção, são descritos os detalhes da implementação do AG.

5.2 Implementação do Método Proposto usando Algoritmos Genéticos

Como mencionado anteriormente, o grande desafio do método proposto é selecionar as posições dos pixels ao longo das direções angular e radial de forma que, para o banco de imagens utilizado, o processamento dos pixels escolhidos deixam o sistema mais confiável.

Para solucionar o problema em questão foi desenvolvido o projeto de um algoritmo genético.

Para a implementação de um AG, é necessário, inicialmente, definir a estrutura dos cromossomos que representarão possíveis soluções para o problema. Foi utilizada uma representação binária cuja dimensão é função das resoluções angular e radial da imagem que é normalizada inicialmente pelo método tradicional de Daugman e que contém todos os possíveis pixels que podem ser selecionados pelo AG. Supõe-se que essas resoluções angular e radial são C_i e L_i respectivamente. A representação do cromossomo é dividida em duas partes, sendo que a primeira delas possui dimensão C_i e fornece informações a respeito da seleção na direção angular, ou seja, identifica de quais linhas radiais (ou, de quais colunas da imagem normalizada) serão amostrados os pixels. A segunda parte do cromossomo possui dimensão L_i e fornece informações a respeito da seleção na direção radial, ou seja, identifica de quais posições em cima das linhas radiais (ou, de quais linhas da imagem normalizada) serão amostrados os pixels. A Figura 5.1 ilustra essa estrutura.

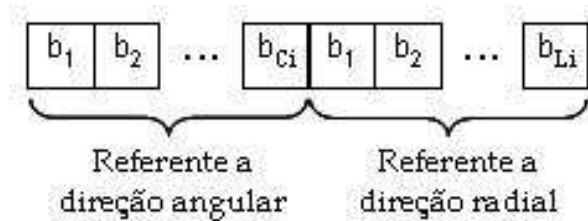


Figura 5.1: *Estrutura do cromossomo utilizado pelo algoritmo genético.*

Cada bit ou, na linguagem de AGs, cada gene do cromossomo, representa uma das possíveis colunas ² ou linhas ³ da imagem normalizada que estão disponíveis para serem escolhidas. Quando o valor de um gene é 1, significa que a linha ou coluna a que ele se refere está sendo selecionada e, quando o

²Se o bit estiver na primeira parte do cromossomo.

³Se o bit estiver na segunda parte do cromossomo.

valor do gene é 0, a linha ou coluna não está sendo selecionada.

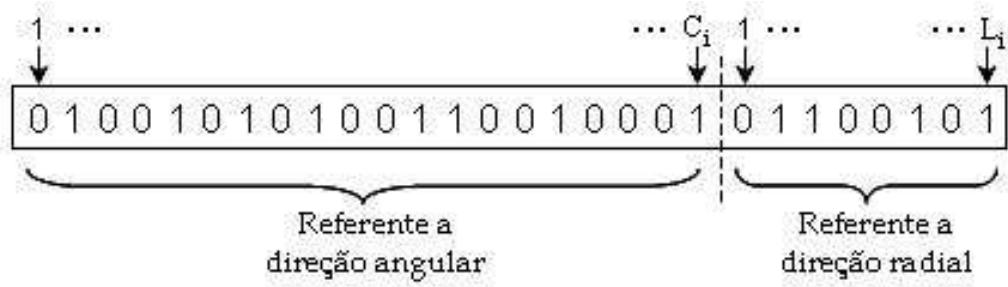
Deseja-se que a solução do AG produza uma representação retangular da íris com determinadas resoluções angular e radial dadas, respectivamente, por C_f e L_f . Para isto, dos C_i bits da primeira parte do cromossomo, C_f bits devem ser iguais a 1 e os demais iguais a 0, e ainda, dos L_i bits da segunda parte do cromossomo, L_f bits devem ser iguais a 1 e os demais iguais a 0. Um cromossomo só é válido se atender a essas condições.

A Figura 5.2 ilustra um exemplo hipotético de uma solução do AG em que $C_i = 20$, $L_i = 8$, $C_f = 8$ e $L_f = 4$.

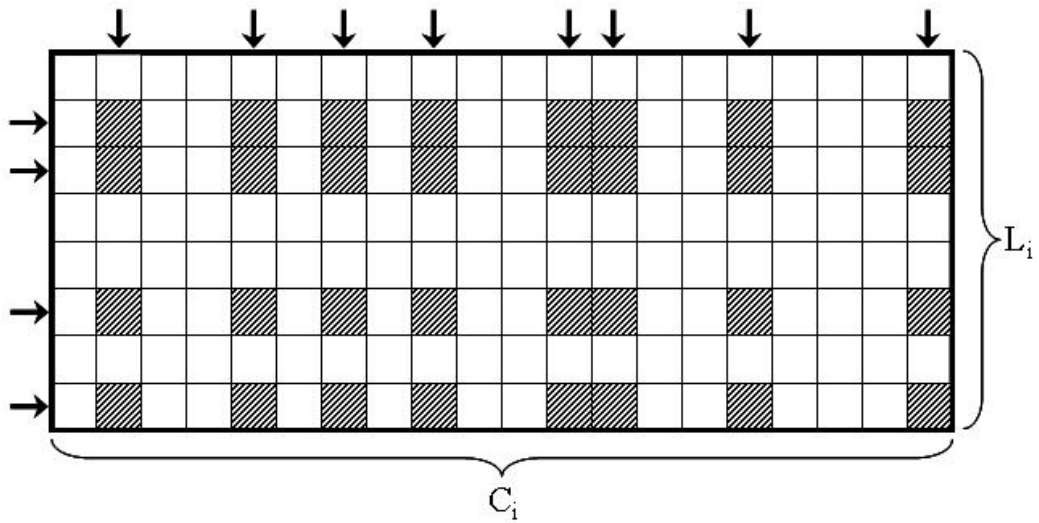
O primeiro passo na execução do AG é a geração da população inicial que contém as possíveis soluções disponíveis para o AG inicialmente. Uma população é formada por cromossomos onde cada cromossomo representa um indivíduo que é uma possível solução para o problema. Assim, é necessário criar cromossomos para se obter a população inicial. A criação dos cromossomos que compõem a população inicial é realizada aleatoriamente, ou seja, o valor binário de cada gene do cromossomo é escolhido ao acaso, porém, deve-se obedecer à premissa de que a primeira parte do cromossomo deve possuir C_f genes iguais a 1, a segunda parte do cromossomo deve possuir L_f genes iguais a 1 e todos os outros genes devem ser iguais a 0.

Uma definição extremamente importante no projeto de um AG é da função de aptidão que atribui uma pontuação (score) para cada indivíduo. Uma medida da eficiência do sistema, a taxa de erro de reconhecimento, foi usada como função de aptidão. O critério usado para definir a função de aptidão é descrita na subseção 5.2.1.

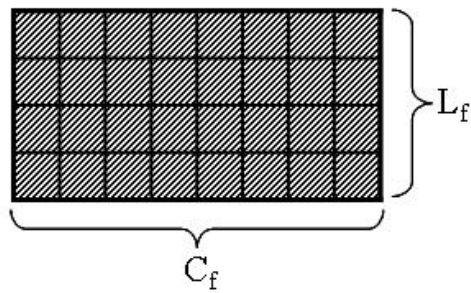
A reprodução é realizada utilizando-se somente um operador de mutação. Diversas pesquisas [20, 26, 27, 28, 66, 69, 70] comprovaram que a mutação é capaz de promover a convergência de uma população em um processo de evolução, mesmo sem a presença do operador de cruzamento (crossover).



(a) Exemplo de um cromossomo (solução do AG).



(b) Resultado da aplicação da solução representada pelo cromossomo acima: pixels amostrados da imagem normalizada gerada inicialmente.



(c) Imagem normalizada formada pelos pixels amostrados.

Figura 5.2: *Exemplo de uma solução do AG.*

A utilização somente da mutação se apresentou como uma alternativa mais interessante para este trabalho, uma vez que se trata de um problema baseado

em ordenação e uma certa aleatoriedade é desejada.

Como tem-se a restrição de que o cromossomo deve possuir sempre uma quantidade fixa de genes iguais a 1 e iguais a 0, um operador genético de mutação adequado para o problema em questão é a mutação trocada (swap mutation) que funciona da seguinte maneira. Primeiro é selecionado um indivíduo da população atual e são escolhidos aleatoriamente dois genes, sendo um igual a 1 e o outro igual a 0. Então esses genes são trocados de lugar originando um descendente válido. Este procedimento é ilustrado na Figura 5.3.

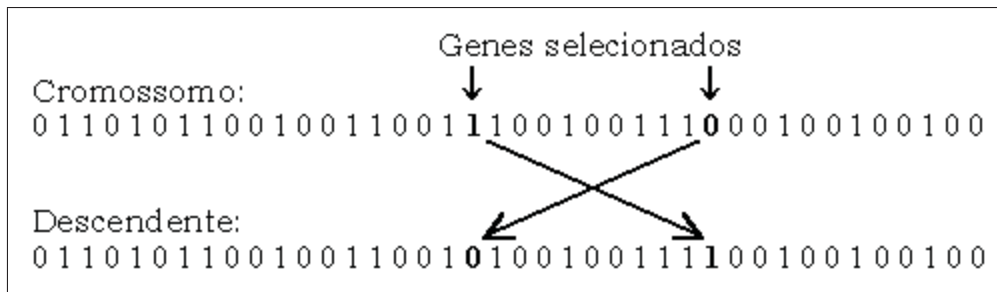


Figura 5.3: *Mutação trocada (Swap mutation).*

Não necessariamente a mutação trocada consiste da troca de apenas um par de genes. Foi implementado um procedimento em que, no momento da mutação de um indivíduo, a quantidade de pares de genes que serão trocados é escolhida aleatoriamente e pode ir desde 1 até um determinado valor máximo previamente especificado.

As duas partes dos cromossomos são processadas independentemente, ou seja, a mutação é aplicada na primeira parte do cromossomo considerando apenas os bits referentes à seleção de pixels na direção angular e depois, é aplicada na segunda parte do cromossomo considerando apenas os bits referentes à seleção de pixels na direção radial. A quantidade máxima de pares que podem ser trocados durante a mutação foi considerada igual a 10 para a primeira parte do cromossomo e igual a 3 para a segunda parte.

Estes valores foram definidos empiricamente. A Figura 5.4 ilustra situações possíveis quando é permitido a troca de até 3 pares.

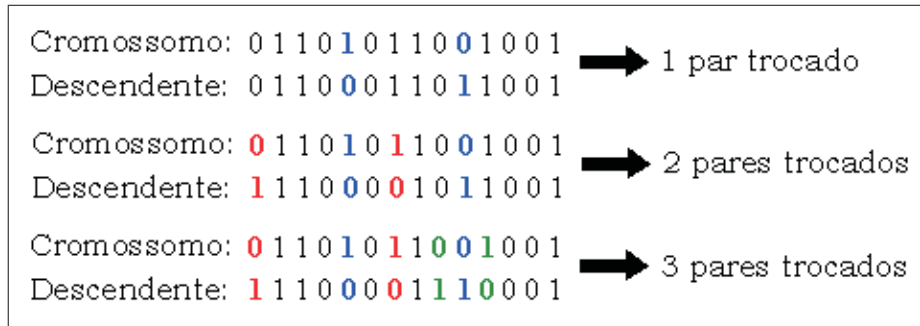


Figura 5.4: *Ilustração da troca de até três pares de genes.*

Os indivíduos que serão submetidos à mutação trocada são selecionados na população corrente pelo “Método da Roleta” [33] que prioriza os indivíduos com melhor aptidão. É selecionada uma quantidade de indivíduos igual ao tamanho da população e todos os indivíduos selecionados sofrem a mutação originando descendentes diferentes.

Pode acontecer do descendente gerado ser menos apto do que seu ancestral. Também é possível que esse mesmo descendente seja mais apto do que um outro indivíduo da população corrente. Assim, para garantir que a próxima população será formada por indivíduos melhores, ou seja, a população converge para um melhor resultado, os descendentes são reunidos com os ancestrais e então os mais aptos são selecionados para formar a próxima geração. Portanto, pode acontecer de um descendente conviver com seu ancestral nessa geração, ou de serem excluídos tanto o descendente quanto seu ancestral. Este procedimento também impede que um indivíduo com um valor de aptidão muito bom seja perdido.

5.2.1 Definição da função de aptidão

A eficiência de um sistema de reconhecimento de íris é avaliada observando a Taxa de Falsa Aceitação (FAR - False Accept Rate) e a Taxa de Falsa Rejeição (FRR - False Reject Rate). Normalmente, se o limiar de decisão para escores de similaridade é variado, a FAR e a FRR variam também. Algumas aplicações de segurança demandam um sistema mais conservativo, com FAR praticamente igual a 0%, enquanto, em outros casos, o processo pode ser mais liberal. Então, esse limiar (*threshold*) é geralmente definido para tornar o sistema mais ou menos conservativo dependendo dos requerimentos da aplicação específica.

Em todos os experimentos deste trabalho, para definir a taxa de erro de reconhecimento, foi escolhido o limiar de decisão que minimiza a soma das taxas FAR e FRR resultantes. Portanto, a taxa de erro de reconhecimento é obtida pela menor soma das taxas FAR e FRR que podem ser obtidas com a simulação do sistema. Então, o valor de aptidão de um indivíduo é dado pelo valor da taxa de erro de reconhecimento obtida quando o sistema é simulado amostrando a combinação de pixels que o indivíduo representa. Isto significa que, para calcular cada valor de aptidão, devem ser realizadas inúmeras comparações intra-classe e inter-classe para que seja possível calcular o erro do sistema.

Deve-se lembrar que, durante a execução do sistema, para a geração dos *templates* utilizados para comparação, as representações retangulares da íris de todas as imagens do banco de dados são geradas pela amostragem dos pixels indicados pelo indivíduo avaliado e depois, essas imagens normalizadas são submetidas aos algoritmos de extração e codificação de características. Quanto menor o valor do erro de reconhecimento para uma determinada combinação de pixels, maior a aptidão do indivíduo e maior a probabilidade

deste ser selecionado para a reprodução.

Note que o método pode ser adaptado para diferentes aplicações uma vez que o critério usado para definir o parâmetro de aptidão pode ser facilmente modificado alterando a definição da chamada taxa de erro de reconhecimento. Por exemplo, para aplicações conservativas, altos valores de aptidão podem ser atribuídos a indivíduos que resultem em uma FAR igual a 0% e uma FRR mínima.

5.3 Resultados Experimentais

Na seção 3.6 foi discutido o comportamento do sistema quando é utilizado o algoritmo tradicional para amostrar os pixels da região da íris. Nesta seção, o método proposto é avaliado pela análise dos efeitos de sua aplicação no que diz respeito à eficiência do sistema. Os resultados obtidos com o novo método serão comparados com os obtidos com o método tradicional.

Em todas as simulações realizadas para avaliar o método proposto, foram executadas as mesmas comparações, com os mesmos parâmetros⁴ e seguindo os mesmos algoritmos daquelas simulações realizadas na seção 3.6.

Para avaliar o método proposto foram realizados diversos experimentos utilizando os bancos de imagens UBIRIS [63] e MMU [2]. Em seguida são apresentados os detalhes dos resultados obtidos com cada um deles.

5.3.1 Resultados com UBIRIS

Quando o banco de imagens UBIRIS foi utilizado na simulação do método tradicional, o sistema apresentou um erro mínimo de 0,07% (ou 99,93% de acerto), lembrando que esta taxa representa a menor soma das taxas FAR e FRR. Este erro mínimo foi obtido quando os pixels foram amostrados uni-

⁴Parâmetros: número de comparações, quantidade de deslocamentos, constantes do filtro Log-Gabor.

formemente para gerar uma representação retangular da íris com resolução 14 x 150, ou seja, resolução radial igual a 14 pixels ⁵ e resolução angular igual a 150 pixels ⁶, como mostrado na Tabela 3.1. Como cada pixel da representação retangular resulta em 2 bits no *template*, são gerados *templates* de 4200 bits (resolução 14 x 300).

No primeiro teste que foi realizado não houve a preocupação em reduzir o tamanho do *template*, pelo contrário, a intenção foi manter a mesma resolução do *template* e verificar se o novo método é capaz de provocar uma melhora na eficiência do sistema. Para executar esse teste foi gerada inicialmente, pelo método tradicional, uma representação retangular da íris com resolução 30 x 250. Esta resolução foi escolhida com a única restrição de ser maior do que 14 x 150. O AG foi aplicado com o objetivo de escolher 14 posições na direção radial (à partir das 30 posições disponíveis) e 150 posições na direção angular (à partir das 250 posições disponíveis). O AG converge para a distribuição de pixels que, quando processados, minimizam o erro do sistema. A Figura 5.5(a) ilustra a melhor solução encontrada onde os pontos brancos representam os pixels selecionados. Esta solução minimiza o erro para 0,0061%. Isto representa que o sistema se torna 0,064% mais eficiente usando o novo método. Uma vez que a confiabilidade de sistemas de reconhecimento de íris é sempre muito alta, esta pequena melhora pode ser considerada muito significativa.

Os demais testes realizados, que são comentados a seguir, procuram investigar a relação custo/benefício, ou seja, a capacidade do método de gerar *templates* menores sem comprometer a confiabilidade do sistema. Para a execução desses testes foram geradas, pelo método tradicional, representações retangulares com resolução 14 x 150 (que foi a que apresentou um melhor

⁵Representação retangular com 14 linhas.

⁶Representação retangular com 150 colunas.

resultado na seção 3.6.1) de onde serão amostrados os pixels pelo AG para a geração de *templates* menores.

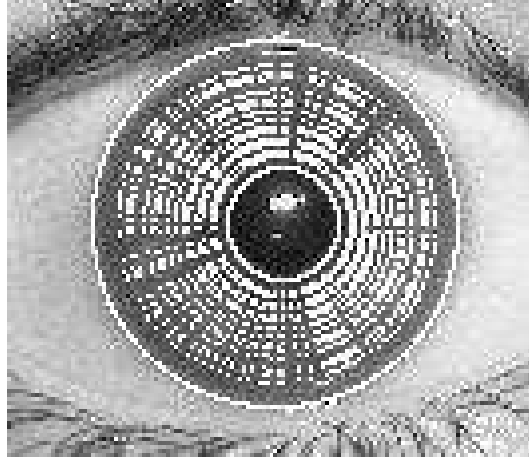
Foram realizados seis testes, sendo que, em cada um deles, o AG foi programado para originar *templates* com tamanhos diferentes. Os valores de resolução da representação retangular utilizados foram 10 x 112, 10 x 54, 7 x 76, 12 x 44, 6 x 60 e 5 x 50, gerando *templates* de 2240, 1080, 1064, 1056, 720 e 500 bits respectivamente. O critério utilizado para a escolha dessas resoluções obedece apenas a restrição de redução do tamanho do *template*. Estes valores de resolução resultam em erros de 0,09%, 0,22%, 0,20%, 0,20%, 0,44% e 1,44%, respectivamente. As Figuras 5.5(b) e 5.5(c) ilustram a melhor solução encontrada em dois casos.

Na Tabela 5.1 são apresentados os parâmetros utilizados para cada simulação e os respectivos resultados obtidos.

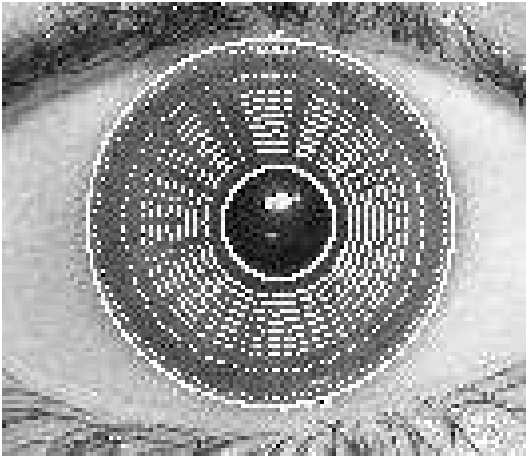
Tabela 5.1: *Resumo dos resultados das simulações utilizando o banco de imagens UBIRIS.*

Resolução da Repres. Retangular	Tamanho do <i>Template</i>	Tamanho da População	Número de Gerações	Menor Erro	Taxa de Acerto
14 x 150 Uniforme	4200 bits	-	-	0,07%	99,93%
14 x 150 Método Novo	4200 bits	70	100	0,012%	99,988%
10 x 112	2240 bits	70	55	0,09%	99,91%
10 x 54	1080 bits	70	300	0,22%	99,78%
7 x 76	1064 bits	70	300	0,20%	99,8%
12 x 44	1056 bits	70	300	0,20%	99,8%
6 x 60	720 bits	70	250	0,44%	99,56%
5 x 50	500 bits	70	250	1,44%	98,56%

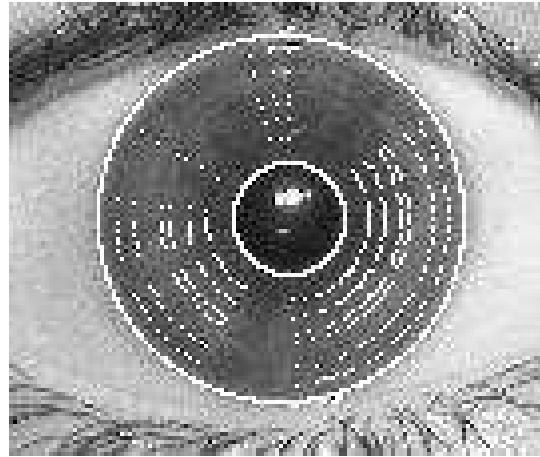
Observa-se que, quando se utiliza a resolução 10 x 112, o tamanho do *template* diminui quase que para a metade enquanto a eficiência do sistema é reduzida em apenas 0,02%.



(a) Melhor solução obtida pelo AG para a resolução 14 x 150.



(b) Melhor solução obtida pelo AG para a resolução 10 x 112.



(c) Melhor solução obtida pelo AG para a resolução 6 x 60.

Figura 5.5: *Soluções do algoritmo genético para o banco de imagens UBIRIS.*

Para as resoluções 10 x 54, 7 x 76 e 12 x 44 o *template* fica com, aproximadamente, 25% do seu tamanho inicial. Como, para essas resoluções, o tamanho do *template* é praticamente o mesmo, as taxas de acerto alcançadas ficam extremamente próximas entre si. Comparando as resoluções 7 x 76 e 12 x 44, a resolução radial aumentou de 7 para 12 (aumento de 71,4%) e a resolução angular diminuiu de 76 para 44 (diminuição de 42,1%), porém, a

taxa de acerto do sistema foi a mesma uma vez que o tamanho do *template* é praticamente o mesmo para os dois casos.

Quando a resolução é igual a 6 x 60, a eficiência do sistema cai 0,37% em relação ao processo tradicional com resolução 14 x 150, em contrapartida, o tamanho do *template* sofre uma redução de quase 6 vezes. Para a resolução igual a 5 x 50, o *template* se torna mais de 8 vezes menor, e, apesar da taxa de acerto ter caído 1,37% em relação ao processo tradicional, ela ainda se encontra em um nível bastante aceitável.

Esses fatores tornam evidente o fato de que a aplicação do método proposto realiza uma seleção mais eficiente das características da íris possibilitando armazenar templates muito menores e ainda assim, conseguir altas taxas de acerto.

O desempenho do reconhecimento biométrico é, geralmente, medido pela geração das chamadas curvas ROC (*Receiver Operating Characteristic*) que plotam a permuta entre as taxas FAR e FRR à medida que o limiar de decisão é variado de conservativo à liberal. É comum tabular pontos específicos dessas curvas, como por exemplo, a FRR quando o limiar de decisão provoca uma FAR igual a 1 em 1000 e o ponto no qual as duas taxas de erro são iguais, $FAR = FRR = EER$ (*Equal Error Rate*).

As curvas ROC resultantes de cada um dos experimentos realizados estão mostrados na Figura 5.6 e os pontos tabulados são apresentados na Tabela 5.2. As coordenadas do gráfico variam somente de 0% até aproximadamente 0,9% para tornar possível a visualização das principais informações de cada curva, permitindo uma comparação visual. Os pontos marcados com um quadrado representam o ponto EER de cada curva.

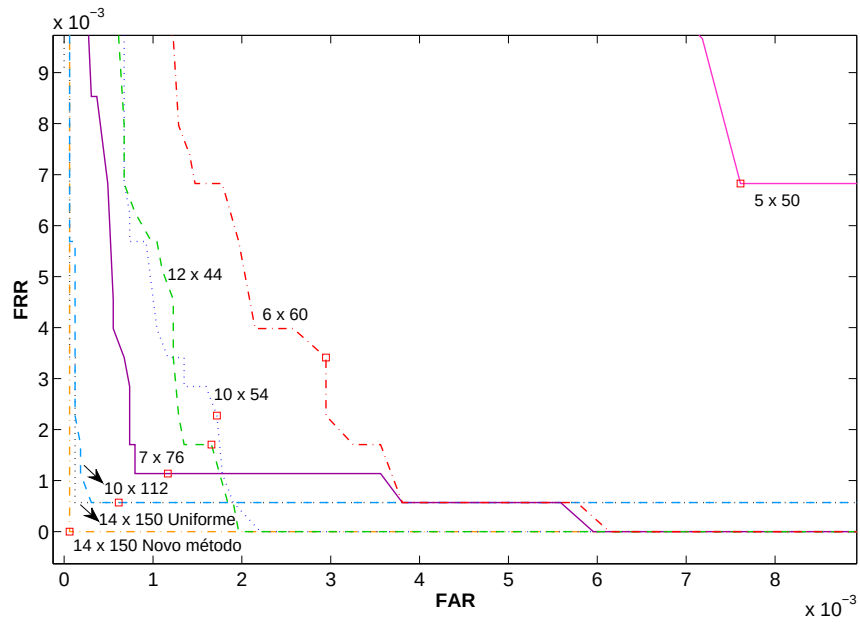


Figura 5.6: *Curvas ROC para o banco de imagens UBIRIS.*

Tabela 5.2: *Tabulações das curvas ROC resultantes dos experimentos utilizando o banco de imagens UBIRIS.*

Resolução da representação retangular	EER	FRR quando FAR=0,001
14 x 150 (Uniforme)	0,000591	0,000569
14 x 150 (Novo Método)	0,000031	0
10 x 112	0,000591	0,000569
10 x 54	0,001997	0,003981
7 x 76	0,001152	0,001137
12 x 44	0,001681	0,005688
6 x 60	0,003179	0,010238
5 x 50	0,007218	0,056880

5.3.2 Resultados com MMU

Com o banco de imagens MMU foram realizados testes similares aos feitos com o banco de imagens UBIRIS.

O melhor resultado obtido com o método tradicional foi um erro de 0,91%

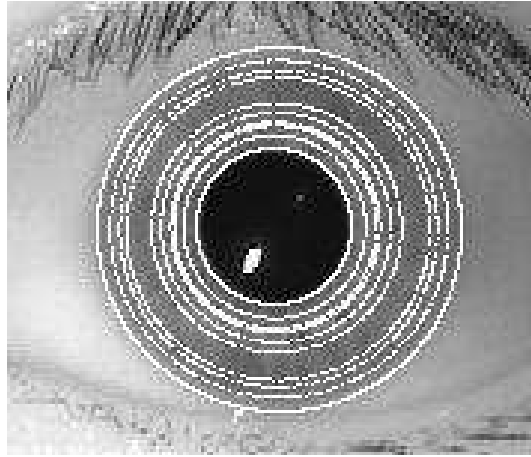
(ou 99,09% de acerto) quando os pixels foram amostrados uniformemente para gerar uma representação retangular da íris com resolução 8 x 370, ou seja, resolução radial igual a 8 pixels e resolução angular igual a 370 pixels, como mostrado na Tabela 3.3. Como cada pixel da representação retangular resulta em 2 bits no *template*, serão gerados *templates* de 5920 bits (resolução 8 x 740).

No primeiro teste foram processados *templates* do mesmo tamanho, porém, provenientes de uma amostragem não uniforme dos pixels na região da íris. Inicialmente, aplicou-se o método tradicional para gerar uma representação retangular com resolução 30 x 500 (resolução qualquer, maior do que 8 x 370) de onde o AG escolheu 8 posições na direção radial (à partir das 30 posições disponíveis) e 370 posições na direção angular (à partir das 500 posições disponíveis). A Figura 5.7(a) ilustra a melhor solução encontrada pelo AG, cujo erro é 0,55%. Observa-se que o sistema ficou 0,36% mais eficiente com o novo método.

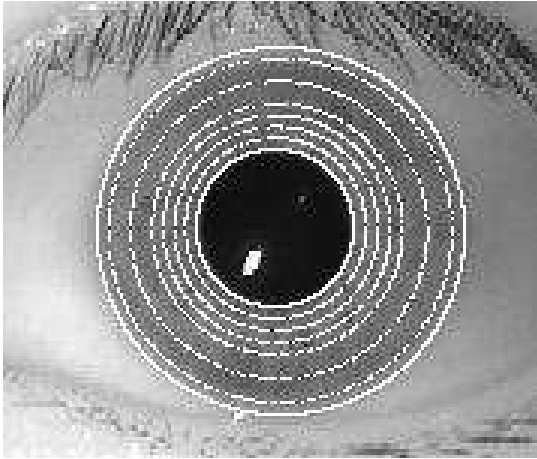
Para realizar os testes processando *templates* menores foram geradas inicialmente, pelo método tradicional, representações retangulares com resolução 8 x 370 (que foi a que apresentou um melhor resultado na seção 3.6.2) de onde serão amostrados os pixels pelo AG.

O sistema foi simulado considerando resoluções da representação retangular iguais a 6 x 276, 4 x 186, 6 x 124, 4 x 94 e 6 x 62 gerando *templates* de 3312, 1488, 1488, 752 e 744 bits respectivamente. A escolha destas resoluções tem como restrição a redução do tamanho do *template*. Estes valores de resolução resultam em erros de 0,81%, 1,24%, 1,59%, 2,78% e 3,44%, respectivamente. As Figuras 5.7(b) e 5.7(c) ilustram a melhor solução encontrada em dois casos.

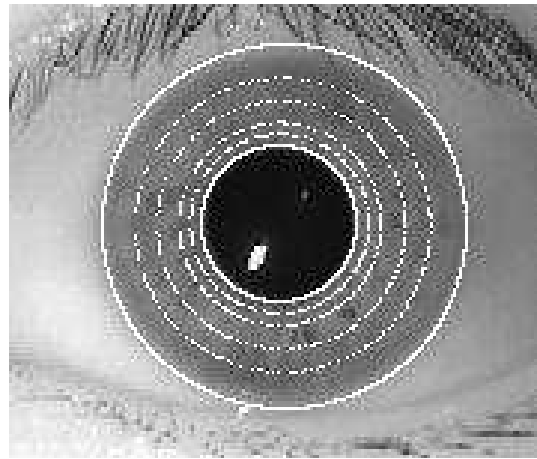
Na Tabela 5.3 são apresentados os parâmetros utilizados para cada simulação e os respectivos resultados obtidos.



(a) Melhor solução obtida pelo AG para a resolução 8×370 .



(b) Melhor solução obtida pelo AG para a resolução 6×276 .



(c) Melhor solução obtida pelo AG para a resolução 4×186 .

Figura 5.7: *Soluções do algoritmo genético para o banco de imagens MMU.*

Observa-se que, utilizando-se a resolução 6×276 , o *template* foi reduzido em mais de 2600 bits e mesmo assim, a eficiência do sistema aumentou em 0,1% em relação ao sistema tradicional.

Para as resoluções 4×94 e 6×62 a queda da taxa de acerto pode ser considerada inexpressiva, diante da vantagem de se reduzir o tamanho do *template* em quase 8 vezes.

A Figura 5.8 mostra as curvas ROC resultantes de todos os experimentos

Tabela 5.3: *Resumo dos resultados das simulações utilizando o banco de imagens MMU.*

Resolução da Repres. Retangular	Tamanho do <i>Template</i>	Tamanho da População	Número de Gerações	Menor Erro	Taxa de Acerto
8 x 370 Uniforme	5920 bits	-	-	0,91%	99,09%
8 x 370 Método Novo	5920 bits	80	300	0,55%	99,45%
6 x 276	3312 bits	80	70	0,81%	99,19%
4 x 186	1488 bits	80	125	1,24%	98,76%
6 x 124	1488 bits	80	290	1,59%	98,41%
4 x 94	752 bits	80	288	2,78%	97,22%
6 x 62	744 bits	80	300	3,44%	96,56%

e a Tabela 5.4 ilustra alguns pontos tabulados. Neste caso, as coordenadas do gráfico variam de 0% até somente 4% para tornar possível a visualização das principais informações de cada curva, permitindo uma comparação visual. Os pontos marcados com um quadrado representam o ponto EER de cada curva.

Tabela 5.4: *Tabulações das curvas ROC resultantes dos experimentos utilizando o banco de imagens MMU.*

Resolução da representação retangular	EER	FRR quando FAR=0,001
8 x 370 (Uniforme)	0,007720	0,017405
8 x 370 (Novo Método)	0,003238	0,022152
6 x 276	0,006176	0,025316
4 x 186	0,006477	0,047468
6 x 124	0,010958	0,041139
4 x 94	0,014046	0,132911
6 x 62	0,018678	0,215190

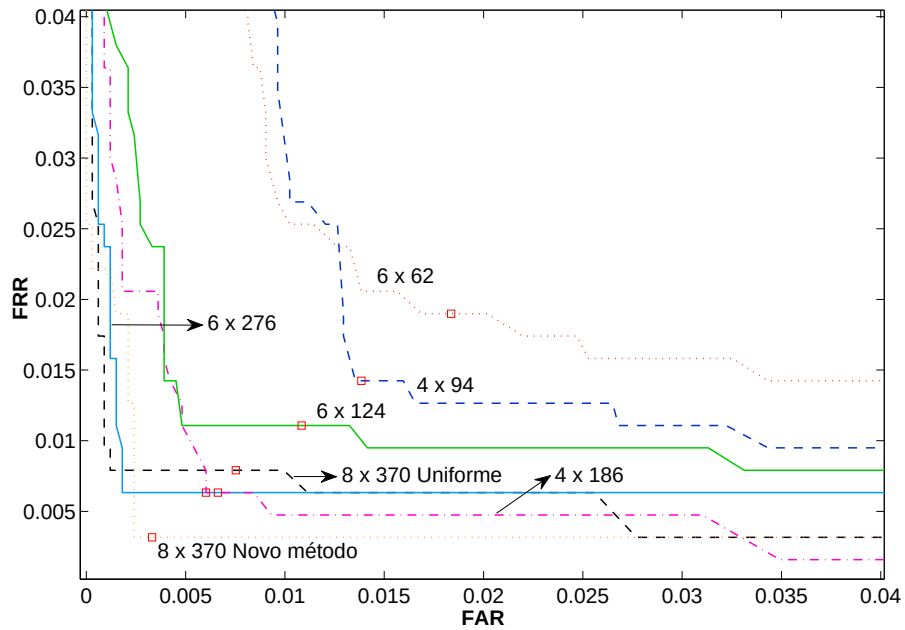


Figura 5.8: *Curvas ROC para o banco de imagens MMU.*

5.4 Conclusões do Capítulo

Neste capítulo foram apresentados os fundamentos do método proposto para realizar uma amostragem não-uniforme dos pixels na região da íris. O método consiste da aplicação de algoritmos genéticos para fazer uma busca pelos pixels que armazenam mais informações da estrutura da íris e que possibilitam uma maior diferenciação de um certo grupo de indivíduos.

Este método foi criado com dois objetivos principais onde o primeiro deles é o aumento da eficiência do sistema e o segundo é possibilitar a geração de *templates* menores mantendo a eficiência do sistema em níveis aceitáveis.

Todos os detalhes da implementação do método foram descritos neste capítulo e, finalmente, foram apresentados os resultados obtidos com as simulações do novo método aplicado a dois bancos de imagens diferentes.

Os resultados mostram que, quando não se tem a preocupação de reduzir o tamanho do *template*, a amostragem não-uniforme implementada foi capaz de aumentar a eficiência do sistema para os dois bancos de imagens utilizados.

Quando o método é aplicado para gerar *templates* menores, a observação dos resultados torna evidente que foi possível reduzir consideravelmente o tamanho do *template*, mantendo altas taxas de acerto. Em aplicações onde a utilização de *templates* menores é um fator impactante, a consequente diminuição na eficiência do sistema pode não ser considerada importante.

Capítulo 6

UTILIZAÇÃO DE ACTIVE SHAPE MODELS PARA SEGMENTAÇÃO DA ÍRIS

Como discutido anteriormente neste trabalho, a tarefa de localizar a porção da imagem que corresponde à região da íris é, tradicionalmente, realizada utilizando métodos de detecção de círculos, uma vez que as bordas que delimitam a íris são aproximadas a círculos. Porém, o contorno externo da íris e o contorno da pupila não são perfeitamente circulares, sendo que, em alguns indivíduos, a íris apresenta um contorno totalmente irregular. Com isso, em alguns casos, a aproximação das bordas da íris a círculos pode ocasionar erros consideráveis.

Outro fator que deve ser levado em consideração é que, algumas vezes, a imagem do olho é capturada com a íris fora de eixo, ou seja, com o indivíduo olhando para uma direção ligeiramente diferente da posição central da câmara. Desta forma, o formato da íris e da pupila se aproxima mais de uma elipse do que de um círculo.

De fato, a performance do sistema de reconhecimento de íris pode ser melhorada consideravelmente se o contorno das bordas da íris forem mais flexíveis e não ficarem limitados ao formato de um círculo.

Muitas vezes a borda externa da íris fica parcialmente escondida atrás das pálpebras, a borda da pupila pode sofrer interferência de reflexos devido à iluminação e ainda, as duas bordas podem ser prejudicadas pela sombra dos cílios. Assim, torna-se necessário ajustar um contorno flexível que pode tolerar interrupções e continuar sua trajetória para formar uma curva fechada [19].

Uma ferramenta adequada para atingir este objetivo é o Active Shape Models (ASM) que modela uma forma flexível de acordo com informações de treinamento, não ficando limitado a uma determinada forma geométrica.

O ASM foi proposto inicialmente por Cootes [72] e pertence a um conjunto de técnicas chamadas de modelos deformáveis. Define-se um modelo deformável como um modelo que, seguindo um critério de otimização, deforma um contorno definido para encontrar um objeto conhecido em uma determinada imagem [31]. Dentre os modelos deformáveis mais conhecidos, podem-se citar Active Contour Models (ou Snakes), Active Appearance Models, Active Blobs e Active Shape Models. A vantagem do ASM é que as instâncias do modelo somente podem ser deformadas das maneiras contempladas no conjunto de treinamento, o que evita variações arbitrárias da forma procurada.

Para segmentar a região da íris em uma imagem, a restrição que o ASM impõe às deformações é desejável, pois, não se espera variações muito significativas dos contornos da borda externa da íris e da borda da pupila uma vez que as possíveis obstruções (por pálpebras, cílios e reflexos) são previstas. Assim, as variações mais típicas da forma podem ser facilmente contempladas no conjunto de treinamento.

No ASM, a forma de um objeto é representada por um conjunto de n pontos que são definidos pelas ordenadas e abscissas do plano da imagem. A quantidade de pontos deve ser pré-definida e suficiente para mostrar todos os detalhes do objeto.

O modelo que é utilizado para descrever a forma e suas variações típicas é baseado na variação da posição de cada ponto nas imagens de treinamento. Cada ponto tem então, uma certa distribuição no espaço da imagem. Para obter um modelo da distribuição dos pontos é necessário que os n pontos da forma sejam marcados manualmente em cada imagem do conjunto de imagens de treinamento. Como a forma varia de uma imagem para a outra, os pontos que definem a forma estarão posicionados em coordenadas diferentes para cada imagem. A análise estatística das coordenadas dos pontos define o ASM.

Com as informações estatísticas provenientes do processo de treinamento e a partir de uma aproximação inicial, uma instância do modelo ASM pode ser ajustada à imagem na qual se busca o objeto descrito pelo modelo. Portanto, a utilização de ASM para localizar a região da íris em uma imagem pode ser dividida em duas etapas principais, que são:

- Etapa de treinamento.
- Etapa de busca da forma em uma imagem.

Neste trabalho foi utilizada uma implementação em MatLab desenvolvida por Ghassan Hamarneh [35] do algoritmo do ASM proposto por Cootes [72]. O programa foi devidamente adaptado para manipular as imagens do banco de imagens de íris UBIRIS [63] e realizar a tarefa de localização da região da íris.

O algoritmo utilizado para implementar cada uma dessas etapas é descrito nas seções 6.1 e 6.2 deste capítulo.

6.1 Etapa de Treinamento

Na técnica do ASM, é necessário que o modelo passe por uma fase de treinamento que consiste em fazer uma análise estatística da posição dos

pontos que são marcados nas imagens de treinamento contornando a forma procurada.

Nesta seção, são apresentados os principais passos do algoritmo proposto por Cootes [72] e desenvolvido por Ghassan Hamarneh [35] para realizar o treinamento do ASM. Nas próximas subseções, são descritos os detalhes do algoritmo.

Os principais passos do algoritmo proposto por Cootes [72] são:

1. Marcar os pontos da forma em todas as imagens do conjunto de treinamento, armazenando as coordenadas deste pontos; (Ver sub-seção 6.1.1)
Para executar este passo é necessário definir como parâmetros de entrada o número de pontos que serão utilizados para definir a forma e ainda, o número de imagens que serão utilizadas para treinamento.
2. Para implementar uma busca multi-resolução deve-se obter as imagens de treino em cada nível de resolução considerado; (Ver sub-seção 6.1.2)
3. Obter o perfil de cada ponto da forma, para cada imagem de treino e para cada nível de resolução; (Ver sub-seção 6.1.3)
4. Obter a estatística do perfil de cada ponto; (Ver sub-seção 6.1.4)
Este passo calcula a média do gradiente normalizado do perfil e sua respectiva matriz de covariância para cada ponto da forma. Estas informações serão úteis na etapa de busca pela forma em uma imagem.
5. Obter a matriz de pesos que é utilizada para aumentar a significância dos pontos da forma cuja posição varia menos de uma imagem de treino para outra. Quanto menor a variação da posição de um ponto, maior o peso atribuído a ele; (Ver sub-seção 6.1.5)
6. Alinhar as formas marcadas em todas as imagens de treino; (Ver sub-seção 6.1.6)

7. Obter a descrição estatística das formas (representadas por conjuntos de pontos) marcadas nas imagens de treino. (Ver sub-seção 6.1.7)

As principais saídas desse passo de processamento são:

- A forma média, que é obtida calculando a média aritmética das coordenadas de todos os pontos das formas de treino;
- Os autovetores; e
- Os autovalores.

Estas saídas serão utilizadas na etapa de busca pela forma em uma imagem.

6.1.1 Marcação das formas nas imagens de treinamento

Como mencionado anteriormente, é necessário que os pontos que definem a forma procurada sejam marcados manualmente em todas as imagens de treinamento. Para isso, deve-se escolher de antemão a quantidade n de pontos que serão utilizados para definir a forma e também quais e quantos desses pontos serão considerados *landmarks* ou “marcos”.

Os landmarks devem ser pontos típicos presentes na forma que são marcados em posições equivalentes em todas as imagens do conjunto de treinamento. Por exemplo, se o objeto tiver a forma retangular, os vértices desse retângulo são pontos típicos, presentes em todas as instâncias desse objeto, sendo portanto os marcos desse objeto [21].

Geralmente, o número de marcos presentes em um objeto não é suficientemente numeroso para definir bem o contorno do objeto. Por isso, devem ser posicionados pontos intermediários entre os marcos.

A região da íris foi representada nas imagens de treinamento utilizando uma única curva fechada definida por pontos distribuídos conforme ilustrado na Figura 6.1.

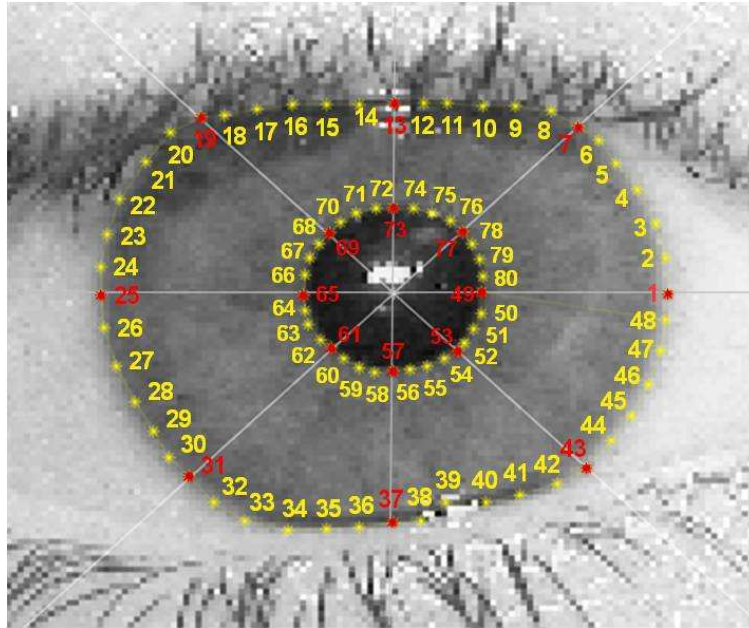


Figura 6.1: *Marcação dos pontos da forma.*

Foram utilizados como pontos de referência (*landmarks*) os pontos posicionados nos ângulos 0° , 45° , 90° , 125° , 180° , 225° , 270° e 315° tanto da borda externa da íris quanto da borda da pupila, totalizando 16 landmarks. Estes pontos estão identificados em vermelho na Figura 6.1.

Para possibilitar a localização com precisão desses landmarks nas imagens de treino foi implementado o seguinte procedimento: na hora de marcar os pontos, o usuário deve, antes de mais nada, selecionar manualmente um ponto que esteja o mais próximo possível do centro da pupila, assim, as quatro retas mostradas na Figura 6.2 são traçadas automaticamente. Essas quatro retas são traçadas passando pelo ponto central da pupila, escolhido pelo usuário, sendo que uma delas está na horizontal, outra na vertical e as outras duas nas diagonais $+45^\circ$ e -45° .

Com as retas sobrepondo a imagem de treino, todos os pontos da forma devem ser selecionados pelo usuário (usando o mouse), na sequência indicada na Figura 6.1, de forma que os landmarks coincidam com as intersecções das

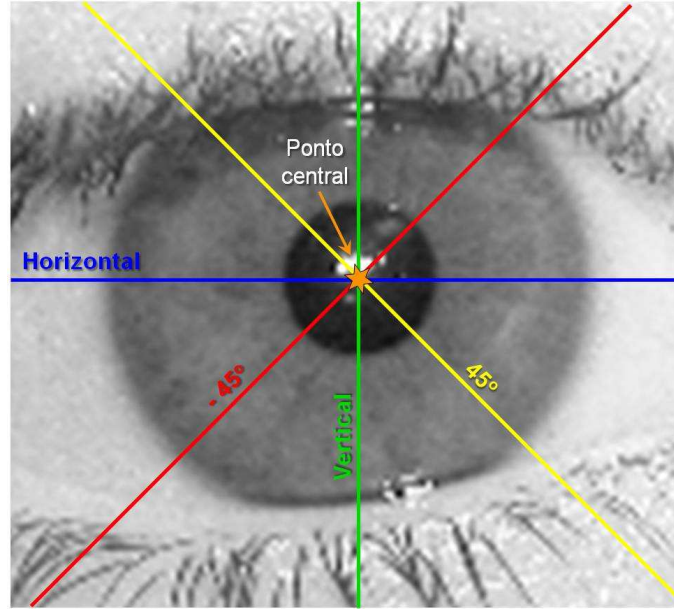


Figura 6.2: *Retas de referência.*

retas com a borda externa da íris e com a borda da pupila.

Como resultado, termina-se com n pontos (cada um identificado por duas coordenadas) para cada uma das N imagens de treinamento. Pode-se dizer também que tem-se N conjuntos de coordenadas para cada ponto da forma. Assim, considerando-se que o j -ésimo conjunto de coordenadas de pontos da i -ésima imagem de treinamento é representado por (x_{ij}, y_{ij}) , o vetor que descreve os n pontos da i -ésima forma no conjunto de treinamento é representado pela Equação 6.1.

$$x_i = [x_{i0}, y_{i0}, x_{i1}, y_{i1}, x_{i2}, y_{i2}, \dots, x_{i(n-1)}, y_{i(n-1)}]^T \quad \text{onde} \quad 1 \leq i \leq N \quad (6.1)$$

Neste trabalho, foi utilizado um total de 80 pontos para definir a forma da região da íris, conforme ilustrado na Figura 6.1. Os primeiros 48 destes pontos contornam a borda externa da íris e os demais 32 pontos contornam a borda da pupila. Foi utilizado para treinamento um conjunto de 100 imagens

provenientes do banco de imagens UBIRIS [63]. Considerando os valores práticos ($n = 80$ e $N = 100$), a Equação 6.1 pode ser escrita como mostrado na Equação 6.2.

$$x_i = [x_{i0}, y_{i0}, x_{i1}, y_{i1}, x_{i2}, y_{i2}, \dots, x_{i(79)}, y_{i(79)}]^T \quad \text{onde} \quad 1 \leq i \leq 100 \quad (6.2)$$

6.1.2 Obtenção das imagens em multi-resolução

Para aumentar a eficiência e a robustez do algoritmo ASM, uma estrutura multi-resolução é implementada. Isto consiste em procurar o objeto primeiro em uma imagem mais grosseira e ir refinando a sua localização em uma série de imagens com melhores resoluções [12].

Para cada imagem de treino é gerada uma pirâmide de imagens como ilustrado na Figura 6.3. Esta pirâmide de imagens contém a mesma imagem, porém, com diferentes resoluções. Na base da pirâmide (Nível 0) tem-se a imagem original. A imagem do nível acima (Nível 1) é formada suavizando-se a imagem original e sub-amostrando-a para obter uma imagem com a metade do número de linhas e a metade do número de colunas. As imagens dos demais níveis são obtidas repetindo o processo de suavização e sub-amostragem da imagem do nível imediatamente anterior.

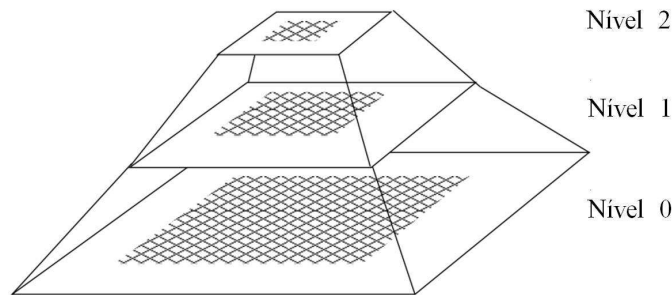


Figura 6.3: Ilustração da pirâmide de imagens obtida por sucessivas suavizações e sub-amostragens de uma imagem [12].

Para fazer a suavização das imagens foi utilizado um filtro gaussiano 5 x 5 proposto por Cootes [12] cuja máscara é mostrada na Figura 6.4.

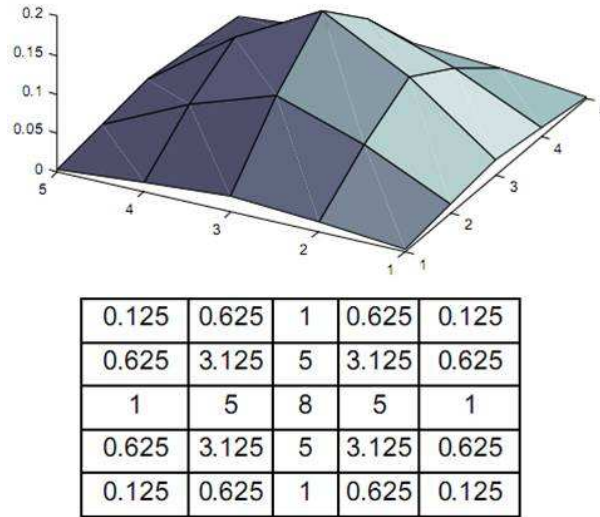


Figura 6.4: *Máscara gaussiana* [12].

A Figura 6.5 ilustra uma imagem nos níveis de resolução de 0 até 3.

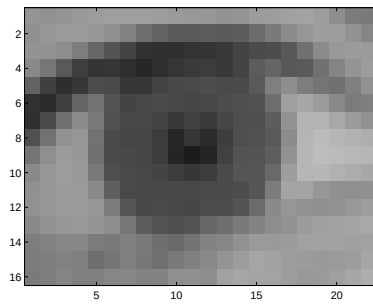
Na implementação deste trabalho foram utilizados apenas os níveis de resolução 0 e 1.

Na seção 6.2.6 é explicado como a estrutura multi-resolução contribui para tornar o sistema mais eficiente e robusto.

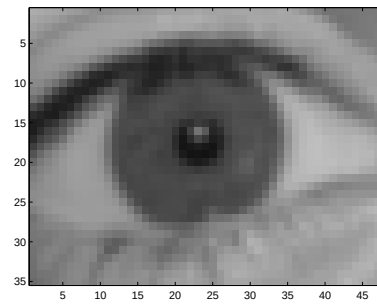
6.1.3 Obtenção do perfil de um ponto da forma

No algoritmo do ASM, o treinamento e a posterior busca da forma em uma nova imagem envolvem analisar as características de variação de intensidade dos pixels próximos aos pontos que definem a forma. Para que essa análise seja possível, é necessário definir quantos e quais pixels próximos aos pontos da forma serão avaliados.

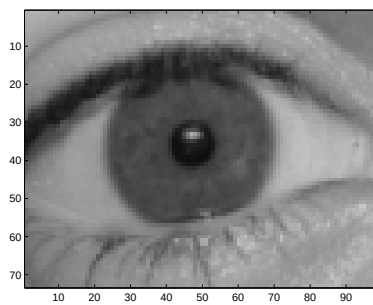
Em geral, qualquer região em torno dos pontos da forma pode ser considerada e estudada. A estratégia mais comum, e a que foi empregada na



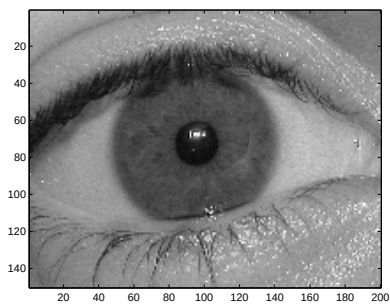
(a) Nível 3



(b) Nível 2



(c) Nível 1



(d) Nível 0

Figura 6.5: Uma imagem em diversos níveis de resolução da pirâmide de imagens.

implementação utilizada neste trabalho, é amostrar uma determinada quantidade de pixels ao longo de uma linha passando por cada ponto da forma e na direção normal ao contorno da forma definida pelo ponto em questão e seus vizinhos. Para obter a direção normal é adotado o seguinte procedimento para cada ponto da forma:

1. Selecionar o ponto anterior e o posterior ao ponto processado no momento;
2. Obter uma reta que passa pelos dois pontos selecionados no item 1;
3. Calcular o ângulo normal à reta obtida no item 2;
4. A partir do ângulo calculado no item 3, obter a reta normal à reta do

item 2 que passa pelo ponto da forma que está sendo processado.

5. Amostrar a intensidade (nível de cinza) de uma determinada quantidade de pixels posicionados ao longo da reta obtida no item 4.

A Figura 6.6 ilustra este procedimento.

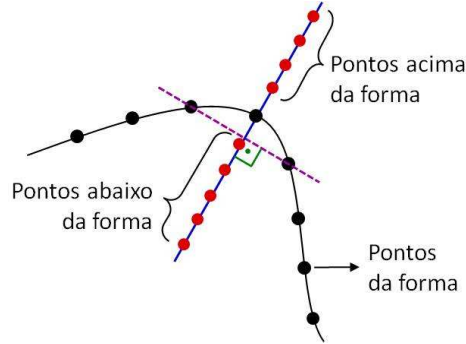


Figura 6.6: *Procedimento para obtenção do perfil dos pontos da forma.*

Para obter o perfil, são amostrados os níveis de cinza de n_{Acima} pontos acima e n_{Abaixo} pontos abaixo do ponto da forma. Assim, a quantidade de valores que formam o perfil de cada ponto da forma é dado por $n_p = n_{Acima} + n_{Abaixo} + 1$.

Na aplicação deste trabalho foram escolhidos empiricamente os valores $n_{Acima} = 5$ e $n_{Abaixo} = 3$. A Figura 6.7 mostra uma imagem (com o nível de resolução original, nível 0) com os pontos selecionados para a obtenção do perfil de cada ponto da forma.

É importante ressaltar que o processo de obtenção do perfil deve ser realizado para todos os níveis de resolução da imagem utilizados.

6.1.4 Obtenção da estatística do perfil

Depois da seleção da intensidade dos pixels que definem o perfil de cada ponto da forma em cada nível de resolução, é necessário obter um modelo estatístico da variação das intensidades desses perfis.

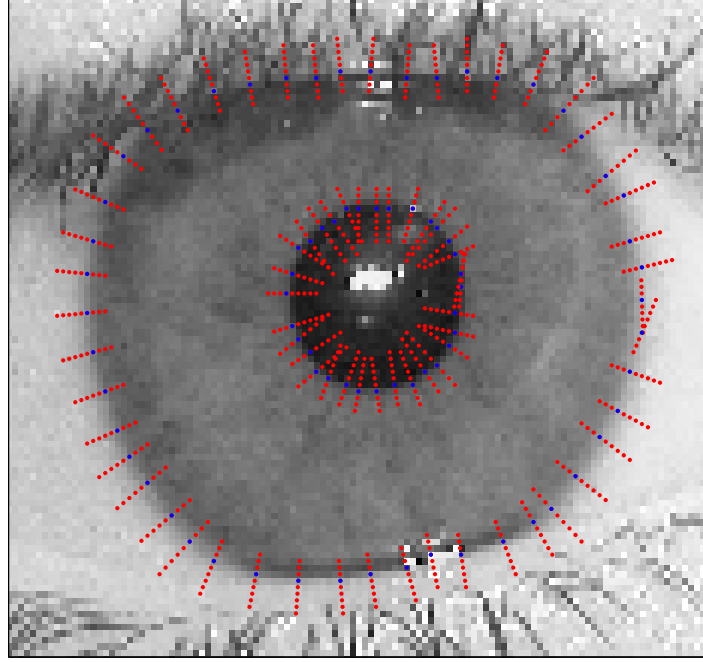


Figura 6.7: *Pontos do perfil de cada ponto da forma.*

Considera-se que, para cada nível de resolução, para cada ponto da forma j da imagem i do conjunto de treinamento, é extraído o perfil de níveis de cinza g_{ij} de comprimento n_p centrado no ponto da forma. Para reduzir o efeito da variação global das intensidades do perfil, é mais conveniente trabalhar com a derivada normalizada dos níveis de cinza do perfil do que com os níveis de cinza propriamente ditos [11].

O perfil de nível de cinza do ponto da forma j da imagem i é o vetor de n_p valores dado pela Equação 6.3.

$$g_{ij} = [g_{ij0} \quad g_{ij1} \quad g_{ij2} \quad \dots \quad g_{ij(n_p-1)}]^T \quad (6.3)$$

A derivada do perfil tem comprimento $n_p - 1$ e é dada pela Equação 6.4.

$$dg_{ij} = [g_{ij1} - g_{ij0} \quad g_{ij2} - g_{ij1} \quad \cdots \quad g_{ij(n_p-1)} - g_{ij(n_p-2)}]^T \quad (6.4)$$

A derivada normalizada do perfil é dada pela Equação 6.5.

$$y_{ij} = \frac{dg_{ij}}{\sum_{k=0}^{n_p-2} |dg_{ijk}|} \quad (6.5)$$

Assim, pela Equação 6.6 pode-se calcular a média da derivada normalizada do perfil para cada ponto da forma j , considerando-se todas as imagens de treino.

$$\bar{y}_j = \frac{1}{N} \sum_{i=1}^N dg_{ij} \quad (6.6)$$

A matriz de covariância da derivada normalizada é dada pela Equação 6.7.

$$C_{y_j} = \frac{1}{N} \sum_{i=1}^N (y_{ij} - \bar{y}_j)(y_{ij} - \bar{y}_j)^T \quad (6.7)$$

A média $\bar{y}_j$ e a matriz de covariância C_{y_j} são utilizadas na etapa de busca da forma em uma imagem.

6.1.5 Obtenção da matriz de pesos

A matriz de pesos é uma matriz diagonal. Os pesos são escolhidos para atribuir maior significância aos pontos da forma que tendem a ser mais estáveis. Um ponto estável apresenta uma posição que varia menos em relação aos demais pontos da forma.

Para calcular esses pesos calcula-se, inicialmente, a distância entre todos os pares de pontos ao longo de toda a forma. Depois, calcula-se a variância de todas essas medidas de distância. Assim, para um determinado ponto, a soma da variância da distância desse ponto a todos os outros mede a instabilidade

desse ponto. Pode-se estimar o peso desse ponto como sendo o inverso desse somatório [35].

Escrevendo matematicamente, suponha que R_{ikl} seja a distância entre os pontos da forma k e l na imagem de treino i . Representando a variância da distância entre os pontos da forma k e l por V_{Rkl} , então o peso para o k -ésimo ponto da forma é dado pela Equação 6.8.

$$w_k = \left(\sum_{l=0}^{n-1} V_{Rkl} \right)^{-1}, \quad \text{onde } 0 \leq k \leq n-1 \quad (6.8)$$

onde n é o número de pontos da forma.

A matriz de pesos é, então, a matriz diagonal mostrada na Equação 6.9.

$$W = \text{diag}(w_1, w_1, w_2, w_2, \dots, w_{n-1}, w_{n-1}) \quad (6.9)$$

6.1.6 Alinhamento das formas do conjunto de treinamento

A marcação das formas nas N imagens de treinamento descrita na seção 6.1.1 resulta em um conjunto de N vetores $\mathbf{x}_i$, $1 \leq i \leq N$. Para estudar a variação da posição de cada ponto da forma no conjunto de imagens de treinamento, todas as formas (cada uma representada por seu correspondente vetor de pontos $\mathbf{x}$) devem ser alinhadas umas às outras.

O alinhamento é realizado buscando os parâmetros de escala, rotação e translação que devem ser aplicados a cada forma para que elas fiquem alinhadas. Matematicamente, para alinhar dois vetores (ou duas formas) $\mathbf{x}_i$ e $\mathbf{x}_k$, deve-se encontrar o fator de escala s , o ângulo de rotação θ e o valor de translação nas duas direções (t_x, t_y) que, quando aplicados a $\mathbf{x}_k$ proporcionam o melhor alinhamento com $\mathbf{x}_i$. A definição de melhor alinhamento é considerada como a transformação que minimiza a soma ponderada dos quadrados das distâncias entre as duas formas [35].

A ponderação é realizada aplicando a matriz de pesos, definida na seção 6.1.5, que é usada para atribuir maior significância aos pontos da forma que tendem a serem mais estáveis.

O algoritmo implementado por Ghassan Hamarneh [35] para alinhar as N formas do conjunto de treinamento é ilustrado no fluxograma da Figura 6.8.

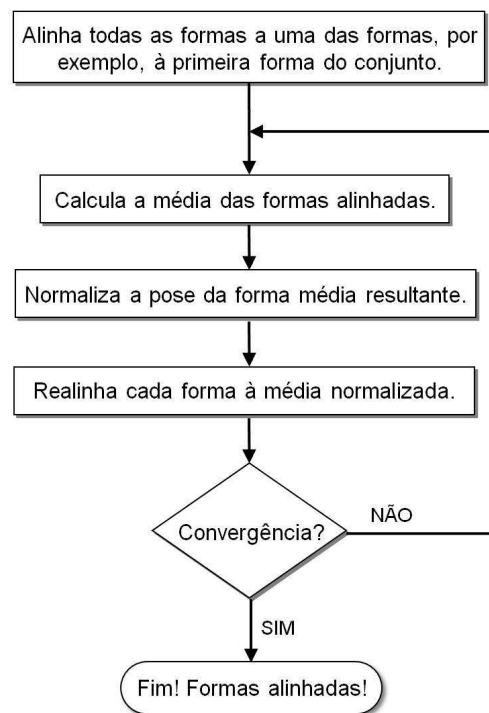


Figura 6.8: *Procedimento para alinhar um conjunto de formas.*

A pose de uma forma é descrita por sua escala, rotação e translação com relação a uma referência conhecida.

A normalização da pose significa:

- a) escalonar a forma de modo que a distância entre dois pontos seja igual a uma certa constante;
- b) rotacionar a forma de modo que a linha que une dois pontos pré-definidos da forma esteja posicionada em uma certa direção; e

c) transladar a forma de modo que ela seja centralizada em uma certa coordenada.

A normalização é realizada com o objetivo de forçar a convergência do processo. Sem ela a forma média pode transladar, expandir ou comprimir indefinidamente.

A convergência é estabelecida se as formas não estiverem sendo modificadas mais do que um limiar pré-definido.

A Figura 6.9 ilustra um conjunto de 30 formas desalinhadas e a Figura 6.10 ilustra o resultado do alinhamento dessas formas.

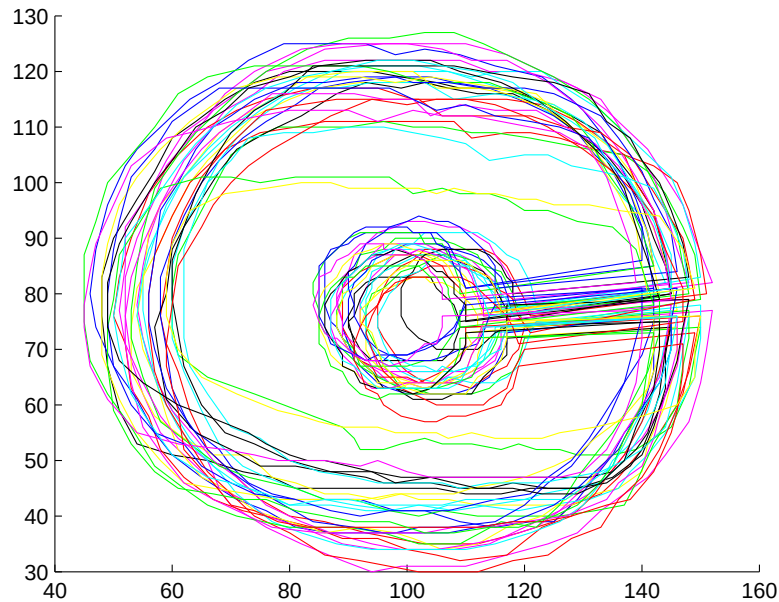


Figura 6.9: *Conjunto de 30 formas desalinhadas.*

6.1.7 Obtenção da estatística das formas

A i -ésima forma alinhada do conjunto de imagens de treinamento é representada pelo vetor $\mathbf{x}_i$, agora contendo novas coordenadas resultantes do

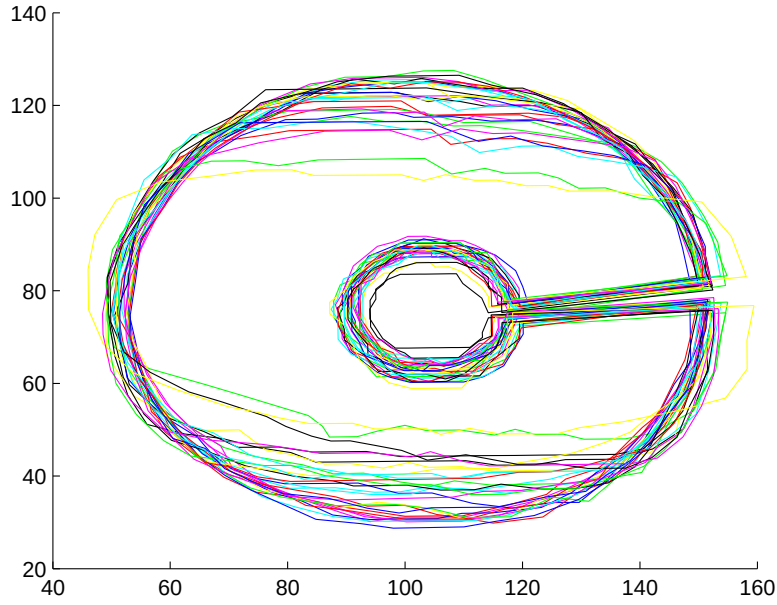


Figura 6.10: *Conjunto de 30 formas alinhadas.*

alinhamento. A dimensão desse é $2n$, então ele pode ser considerado como um ponto em um espaço de dimensão $2n$ ($2n$ -D). Os N vetores representando as N formas alinhadas mapeiam uma nuvem de N pontos no mesmo espaço $2n$ -D. Considera-se que esses N pontos estão contidos em uma região desse espaço $2n$ -D chamada de Domínio de Formas Aceitável (ASD - Allowable Shape Domain). Todos os pontos nessa região originam uma forma similar às outras formas do ASD.

A medida de similaridade de duas formas é a seguinte: quanto menor a distância Euclidiana entre dois pontos (representando duas formas) no espaço $2n$ -D, mais similares são as formas. A distância Euclidiana d entre os dois pontos representando as duas formas $\mathbf{x}_i$ e $\mathbf{x}_k$ é dada na Equação 6.10.

$$d_{ik} = \sqrt{(x_{i0} - x_{k0})^2 + (y_{i0} - y_{k0})^2 + (x_{i1} - x_{k1})^2 + (y_{i1} - y_{k1})^2 + \dots \\ \dots + (x_{i(n-1)} - x_{k(n-1)})^2 + (y_{i(n-1)} - y_{k(n-1)})^2}$$

ou

$$d_{ik}^2 = (\mathbf{x}_i - \mathbf{x}_k)^T (\mathbf{x}_i - \mathbf{x}_k) \quad (6.10)$$

A Equação 6.11 mostra o cálculo de d_{ik} ao considerar a ponderação obtida ao utilizar a matriz de pesos $\mathbf{W}$ com o objetivo de atribuir maior importância aos pontos da forma que variam menos.

$$d_{ik} = \sqrt{w_0(x_{i0} - x_{k0})^2 + w_0(y_{i0} - y_{k0})^2 + w_1(x_{i1} - x_{k1})^2 + w_1(y_{i1} - y_{k1})^2 + \dots \\ \dots + w_{n-1}(x_{i(n-1)} - x_{k(n-1)})^2 + w_{n-1}(y_{i(n-1)} - y_{k(n-1)})^2}$$

ou

$$d_{ik}^2 = (\mathbf{x}_i - \mathbf{x}_k)^T \mathbf{W} (\mathbf{x}_i - \mathbf{x}_k) \quad (6.11)$$

onde

$$\mathbf{x}_i = [x_{i0}, y_{i0}, x_{i1}, y_{i1}, \dots, x_{i(n-1)}, y_{i(n-1)}]^T$$

$$\mathbf{W} = \text{diag}(w_0, w_0, w_1, w_1, \dots, w_{n-1}, w_{n-1})$$

Agora, pretende-se encontrar as principais características que influenciam o comportamento da variação dos N pontos no espaço $2n$ -D definido pelas $2n$ variáveis de $\mathbf{x}$. Deseja-se, ainda, diminuir a quantidade de variáveis necessárias para representar as variações das formas. Para isso foi utilizado o chamado *Principal Component Analysis* (PCA) com o qual pode-se gerar um novo conjunto de variáveis chamadas componentes principais. Cada componente principal é uma combinação linear das variáveis originais [49]. Todas

as componentes principais são ortogonais entre si, portanto, não existem informações redundantes. As componentes principais como um todo formam uma base ortogonal para o espaço de dados.

A primeira componente principal é um único eixo no espaço. Quando se faz a projeção de cada observação (N vetores representando as formas) nesse eixo, os valores resultantes formam uma nova variável. A variância dessa variável é máxima dentre todas as possíveis escolhas do primeiro eixo, isto é, representa a máxima variação da forma.

A segunda componente é outro eixo no espaço, perpendicular ao primeiro. Projetando as N observações nesse eixo, gera-se outra nova variável. A variância desta variável é máxima dentre todas as possíveis escolhas do segundo eixo. A dimensão do conjunto completo de componentes principais é a mesma do conjunto de variáveis original ($2n$).

Em muitas aplicações, pode-se assumir que as primeiras componentes principais descrevem uma alta porcentagem da variância total da informação original. Com isso, a dimensão do modelo pode ser reduzida e as variações das formas podem ser descritas com uma menor quantidade de variáveis (menor do que $2n$).

Uma forma de se obter as componentes principais é usando uma decomposição de autovalores da matriz de covariância da matriz de observação [10]. A matriz de observação contém m linhas de observações e n colunas de variáveis. Para o caso deste trabalho em específico, tem-se N observações (vetores representando formas) e $2n$ variáveis (as coordenadas (x, y) de cada ponto da forma).

É possível expressar cada ponto da forma como uma combinação linear das componentes principais. Além disso, pode-se expressar a diferença entre cada vetor e a média de todos os vetores como uma combinação linear das componentes principais, uma vez que esse vetor diferença também cairá no

espaço $2n$ -D definido pelas componentes principais.

Denotando o vetor média por $\bar{\mathbf{x}}$ e o vetor diferença entre o vetor $\mathbf{x}_i$ e $\bar{\mathbf{x}}$ por $d\mathbf{x}_i$ pode-se escrever a Equação 6.12.

$$d\mathbf{x}_i = \mathbf{x}_i - \bar{\mathbf{x}} \quad (6.12)$$

A forma média é obtida utilizando-se a Equação 6.13.

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \quad (6.13)$$

A matriz de covariância para os pontos das formas é mostrada na Equação 6.14.

$$C_x = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \quad (6.14)$$

O autovetor unitário (p_l , $0 \leq l \leq 2n - 1$) de C_x é obtido utilizando-se a Equação 6.15.

$$C_x p_l = \lambda_l p_l \quad (6.15)$$

onde λ_l é o l -ésimo autovalor de C_x , $\lambda_l \geq \lambda_{l+1}$ e $p_l^T p_l = 1$.

Representando a diferença $d\mathbf{x}_i$ como uma combinação linear das componentes principais tem-se a Equação 6.16.

$$d\mathbf{x}_i = b_{i0}p_0 + b_{i1}p_1 + \dots + b_{i(2n-1)}p_{(2n-1)} \quad (6.16)$$

onde b_{il} é um escalar que pondera p_l para a i -ésima forma e p_l é o l -ésimo eixo de componente principal ou vetor coluna, normalizado para possuir comprimento unitário, isto é, $p_l^T p_l = 1$.

Como as componentes principais são mutuamente ortogonais, elas também são ortonormais como mostrado na Equação 6.17.

$$p_l^T p_m = \begin{cases} 1 & l = m \\ 0 & l \neq m \end{cases} \quad (6.17)$$

Equivalentemente, pode-se escrever $\mathbf{x}_i = \bar{\mathbf{x}} + d\mathbf{x}_i$ e reescrever $d\mathbf{x}_i$ como na Equação 6.18.

$$d\mathbf{x}_i = b_{i0}p_0 + b_{i1}p_1 + \dots + b_{i(2n-1)}p_{(2n-1)} = \mathbf{P}\mathbf{b}_i \quad (6.18)$$

onde $\mathbf{b}_i = [b_{i0} \quad b_{i1} \quad \dots \quad b_{i(2n-1)}]^T$ e $\mathbf{P} = [p_0 \quad p_1 \quad \dots \quad p_{(2n-1)}]^T$

Assim, x_i pode ser obtido pela Equação 6.19.

$$\mathbf{x}_i = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}_i \quad (6.19)$$

e $\mathbf{b}_i$ pode ser encontrado pela Equação 6.20.

$$\mathbf{b}_i = \mathbf{P}^{-1}(\mathbf{x}_i - \bar{\mathbf{x}}) \quad (6.20)$$

Uma vez que $\mathbf{P}$ é uma matriz ortogonal (matriz quadrada com colunas ortonormais) [71], tem-se $\mathbf{P}^T\mathbf{P} = \mathbf{P}\mathbf{P}^T$ e $\mathbf{b}_i$ é encontrado utilizando-se a Equação 6.21.

$$\mathbf{b}_i = \mathbf{P}^T(\mathbf{x}_i - \bar{\mathbf{x}}) \quad (6.21)$$

Resumindo, tem-se N vetores que representam formas, tendo uma forma média $\bar{\mathbf{x}}$. Cada vetor pode ser expresso como uma soma da forma média e uma soma ponderada das componentes principais.

Definindo os pesos como mostrado na Equação 6.21 com $1 \leq i \leq N$, o resultado é uma forma $\mathbf{x}_i$. Porém, deve-se definir outros pesos como $\mathbf{b} = \mathbf{P}^T(\mathbf{x} - \bar{\mathbf{x}})$ onde $x \notin \{x_1 \quad x_2 \quad \dots \quad x_N\}$, resultando em uma forma que não está no conjunto de treinamento. Para determinar o quanto essa nova forma é similar às formas do conjunto de treinamento, calcula-se a distância (d_{ik})

entre elas.

Se é desejado que o vetor de pesos $\mathbf{b}$ seja escolhido de modo que a forma resultante seja considerada uma forma aceitável ou permitida do objeto de procura, então, é necessário impor alguns limites a esses pesos. Limitando b_l a $b_{l_{min}} \leq b_l \leq b_{l_{max}}$ para $0 \leq l \leq 2n - 1$, é adequado escolher $b_{l_{min}} = -b_{l_{max}}$ sendo que $b_{l_{max}}$ deve ser proporcional à variância da projeção do conjunto de treinamento ao longo da l -ésima componente principal.

Relembrando o objetivo dessa análise que é reduzir a dimensionalidade das informações originais e descrever as variações das formas com um menor número de variáveis, agora é possível expressar as N formas, tendo uma média $\bar{\mathbf{x}}$, como a soma da forma média com uma soma ponderada de algumas componentes principais (não sendo necessária todas). Assume-se que as primeiras t (de um total de $2n$) componentes principais correspondem a uma porcentagem suficientemente alta da variância total dos dados originais. A variância total V_T desse conjunto é obtida pela soma de todos os autovalores, ou seja, $V_T = \sum_{l=0}^{2n-1} \lambda_l$.

Deve-se definir a porcentagem f_v da variância do conjunto de treinamento que se deseja preservar. Assim, o número t de autovetores a serem mantidos é definido como o menor valor t que satisfaça a inequação 6.22.

$$\sum_{l=0}^{t-1} \lambda_l \geq f_v V_T \quad (6.22)$$

Neste trabalho a proporção da variância expressa pelo modelo foi definida empiricamente como $f_v = 0,5$ (50%).

Enfim, tem-se a Equação 6.23.

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b} \quad (6.23)$$

onde $\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$ e considerando apenas t componentes principais tem-

se $\mathbf{b} = [b_0 \ b_1 \ \dots \ b_{t-1}]^T$ e $\mathbf{P} = [p_0 \ p_1 \ \dots \ p_{t-1}]^T$.

Os elementos de $\mathbf{b}$ são restritos para permanecerem na faixa $\pm m\sqrt{\lambda_l}$, para $0 \leq l \leq t-1$. A constante m geralmente assume um valor entre 2 e 3 [21]. Essa restrição dos valores dos elementos de $\mathbf{b}$ limita a deformação do contorno do objeto em uma proporção da variância dos pontos.

6.2 Etapa de Busca da Forma em uma Imagem

Na etapa de treinamento, dado um conjunto de formas devidamente alinhadas, representando as variações de uma classe de objetos, pôde-se modelar essas formas por $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}$, onde $\bar{\mathbf{x}}$ é a forma média, $\mathbf{b}$ é um vetor de pesos e $\mathbf{P}$ é a matriz de autovetores contendo as t primeiras componentes principais que descrevem a maior parte das variações das formas do conjunto de treinamento. Modificado $\mathbf{b}$ pode-se gerar novas formas.

Agora, pretende-se estudar meios para encontrar uma instância da classe de objetos em imagens que não fazem parte do conjunto de treinamento. Uma forma inicial é forçada a se deformar em um processo iterativo até que ocorra uma maior correspondência entre os dados da imagem e o modelo.

Nesta seção, são apresentados os principais passos do algoritmo desenvolvido por Ghassan Hamarneh [35] para realizar a busca de uma forma em uma imagem.

Os principais passos do algoritmo são:

1. Estimar uma forma inicial. (Ver sub-seção 6.2.1)
2. Calcular o deslocamento $d\mathbf{x}_i$ que cada ponto da forma deve sofrer para tentar se ajustar ao objeto da imagem que está sendo processada. (Ver sub-seção 6.2.2)

Examinando a região da imagem em torno de cada ponto da forma $\mathbf{x}_i$ uma nova localização para esses pontos deve ser obtida como $\mathbf{x}_i + d\mathbf{x}_i$.

3. Encontrar os parâmetros para modificar a pose da forma. (Ver sub-seção 6.2.3)

Para modificar a pose deve-se aplicar à forma as transformações de escala, rotação e translação de modo a mover $\mathbf{x}_i$ para o mais próximo possível de $\mathbf{x}_i + d\mathbf{x}_i$. Portanto, neste passo, tem-se como objetivo obter os parâmetros de escala, rotação e translação que otimizam essa aproximação.

4. Encontrar os parâmetros para deformar o contorno da forma. (Ver sub-seção 6.2.4)

O contorno da forma é deformado alterando os peso das componentes principais, ou seja, alterando o valor dos elementos do vetor $\mathbf{b}$. Essa deformação é realizada no sentido de tornar $\mathbf{x}_i$ o mais parecido possível com $\mathbf{x}_i + d\mathbf{x}_i$.

5. Aplicar os parâmetro encontrados para a modificação da pose e para a deformação da forma. (Ver sub-seção 6.2.5)

Como resultado tem-se uma nova forma que está mais ajustada ao objeto procurado na imagem.

6. Se o número máximo de iterações (que deve ser pré-estabelecido) não tiver sido alcançado, voltar ao passo 2.

6.2.1 Estimativa da forma inicial

Assume-se que uma instância de um objeto é descrita pela soma da forma média obtida no treinamento com a soma ponderada das componentes principais, sendo que esta soma tem a possibilidade de ser transladada, rotacionada

e escalonada. Isto significa que pode-se expressar a estimativa da forma inicial $\mathbf{x}_i$ como uma versão escalonada, rotacionada e transladada da forma de referência $\mathbf{x}_l$ como mostrado na Equação 6.24.

$$\mathbf{x}_i = M(s_i, \theta_i)[\mathbf{x}_l] + t_i \quad (6.24)$$

onde

$$M(s_i, \theta_i) = s_i \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix}$$

e

$$\mathbf{t}_i = [t_{xi} \quad t_{yi} \quad t_{xi} \quad t_{yi} \quad \dots \quad t_{xi} \quad t_{yi}]^T$$

A matriz $M(s_i, \theta_i)$ escalona a forma de s_i e a rotaciona de θ_i , enquanto o vetor $\mathbf{t}_i$ translada a forma nas direções x e y .

A forma $\mathbf{x}_l$ pode ser expressa como $\mathbf{x}_l = \bar{\mathbf{x}} + d\mathbf{x}_l$, onde $d\mathbf{x}_l = \mathbf{P}\mathbf{b}_l$. Então, a estimativa inicial pode ser escrita na forma da Equação 6.25.

$$\mathbf{x}_i = M(s_i, \theta_i)[\bar{\mathbf{x}} + d\mathbf{x}_l] + t_i \quad (6.25)$$

6.2.2 Obtenção do deslocamento desejado

O deslocamento ($d\mathbf{x}_i$) que deve ser aplicado a cada ponto da forma, para que ela se ajuste melhor ao objeto procurado na imagem, é calculado usando as informações de estatística do perfil que foram obtidas na etapa de treinamento [35].

O mesmo procedimento descrito na sub-seção 6.1.3 para obter o perfil de cada ponto das formas de treinamento é executado novamente para definir uma região de busca em torno de cada ponto da forma que será deformada. Portanto, define-se uma linha passando pelo ponto da forma e perpendicular ao contorno formado pelo ponto e seus vizinhos (da mesma forma como ilustrado na Figura 6.6) e uma determinada quantidade n_s de pontos ao longo

dessa linha são considerados para formar o perfil de busca. A quantidade de pontos do perfil de busca (n_s) deve ser maior do que a quantidade de pontos do perfil definido na etapa de treinamento (representada por n_p na sub-seção 6.1.3).

Ao longo do perfil de busca de cada ponto da forma, deve-se procurar por um sub-perfil (de comprimento n_p) que tenha características similares àsquelas obtidas no treinamento. Para isso deve-se coletar os níveis de intensidade dos pixels ao longo do perfil de busca, obter a derivada e normalizá-la da mesma forma como foi executado na etapa de treinamento (Sub-seção 6.1.4) para obter a derivada normalizada do perfil. Finalmente, procura-se ao longo da derivada normalizada do perfil de busca pelo sub-perfil que apresenta o melhor casamento com a média da derivada normalizada do perfil obtida no treinamento.

O perfil de busca $\mathbf{s}_i$ ao longo do ponto i da forma é dado por

$$\mathbf{s}_i = [s_{i0} \quad s_{i1} \quad s_{i2} \quad \dots \quad s_{i(n_s-1)}]^T$$

A derivada do perfil de busca para o ponto i da forma tem comprimento $n_s - 1$ e é dada por

$$d\mathbf{s}_i = [s_{i1} - s_{i0} \quad s_{i2} - s_{i1} \quad s_{i3} - s_{i2} \quad \dots \quad s_{i(n_s-1)} - s_{i(n_s-2)}]^T$$

Assim, a derivada normalizada do perfil de busca é obtida utilizando-se a Equação 6.26.

$$y_{si} = \frac{ds_i}{\sum_{k=0}^{n_s-2} |ds_{ik}|} \quad (6.26)$$

Agora, busca-se em y_{si} por um sub-perfil que casa com $\bar{y}_i$ (média da derivada normalizada do perfil obtida no treinamento). Denotando o sub-intervalo de y_{si} centrado no d -ésimo pixel de y_{si} por $\mathbf{h}(d)$, deve-se encontrar o valor de d que torna o sub-intervalo $\mathbf{h}(d)$ o mais similar possível a $\bar{y}_i$. Isto pode ser realizado definindo a função erro quadrático (que decresce a medida

que o ajuste fica melhor) e calculando o valor de d que a minimiza. A função erro quadrático é obtida utilizando-se a Equação 6.27.

$$f(d) = (\mathbf{h}(d) - \bar{y}_i)^T (\mathbf{h}(d) - \bar{y}_i) \quad (6.27)$$

Tem-se a possibilidade de ponderar o erro pela inversa da matriz de covariância de y_i ($\mathbf{C}_{y_i}^{-1}$) originando a Equação 6.28 que é conhecida como distância de Mahalanobis..

$$f(d) = (\mathbf{h}(d) - \bar{y}_i)^T \mathbf{C}_{y_i}^{-1} (\mathbf{h}(d) - \bar{y}_i) \quad (6.28)$$

Este processo de minimização da distância de Mahalanobis é ilustrado na Figura 6.11.

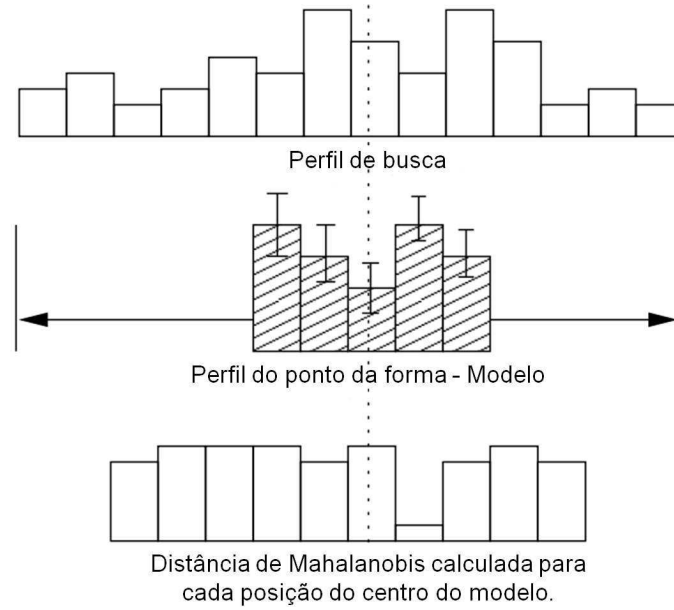


Figura 6.11: Procura ao longo do perfil de busca pelo sub-perfil que fornece o melhor casamento com o perfil da forma [10].

Deste modo a posição para a qual o ponto i da forma deve se mover fica determinada, sendo possível então, calcular qual será o seu deslocamento $d\mathbf{x}_i$. O mesmo procedimento é repetido para todos os pontos da forma para

se obter um vetor com todas as sugestões de deslocamento ($d\mathbf{x}_i$ para $0 \leq i \leq n - 1$).

6.2.3 Obtenção dos parâmetros para modificar a pose da forma

A partir da estimativa inicial da forma pretende-se ajustá-la ao objeto na imagem de busca. Examinando a região da imagem em torno de cada ponto da forma armazenado no vetor $\mathbf{x}_i$, calculou-se na sub-seção 6.2.2 a nova localização desejada para esses pontos: $\mathbf{x}_i + d\mathbf{x}_i$.

É necessário ajustar os parâmetros da pose da forma (escala, rotação e translação) e também os parâmetros de deformação do contorno da forma (pesos das componentes principais $\mathbf{b}$) para mover a estimativa atual da forma $\mathbf{x}_i$ para o mais próximo possível de $\mathbf{x}_i + d\mathbf{x}_i$ ao mesmo tempo que satisfaz as limitações impostas para produzir formas aceitáveis e permitidas.

Obter os parâmetros da pose consiste em encontrar o adicional de escala ($1 + ds$), de rotação ($d\theta$) e de translação (dt_x, dt_y) que deve ser aplicado ao vetor $\mathbf{x}_i$ para alinhá-lo ao vetor $\mathbf{x}_i + d\mathbf{x}_i$. Este alinhamento é realizado usando o mesmo algoritmo descrito na sub-seção 6.1.6. A Equação 6.29 representa matematicamente esta idéia.

$$\mathbf{x}_i = M(s_i, \theta_i)[\mathbf{x}_l] + \mathbf{t}_i \xrightarrow{(1+ds), d\theta, dt} \mathbf{x}_i + d\mathbf{x}_i$$

ou

$$M(s_i(1 + ds), \theta_i + d\theta)[\mathbf{x}_l] + \mathbf{t}_i + d\mathbf{t} \longrightarrow \mathbf{x}_i + d\mathbf{x}_i \quad (6.29)$$

É importante salientar que a variação da pose da forma não é capaz de realizar um ajuste perfeito pois, para isso, também é necessário provocar deformações no contorno da forma como é discutido na sub-seção 6.2.4.

6.2.4 Obtenção dos parâmetros para deformar o contorno da forma

Inicialmente foi realizado o alinhamento da forma inicial $\mathbf{x}_i$ à forma desejada $\mathbf{x}_i + d\mathbf{x}_i$, obtendo-se os parâmetros de variação da pose $((1 + ds), d\theta$ e $d\mathbf{t})$. Porém, ainda existem ajustes residuais que só podem ser conseguidos deformando o contorno da forma. Deseja-se aqui, calcular o ajuste $d\mathbf{x}$ (a ser adicionado a $\mathbf{x}_l$) necessário para fazer $\mathbf{x}_i$ se mover para $\mathbf{x}_i + d\mathbf{x}_i$ quando combinado com as transformações de escala, rotação e translação [10].

Conhecendo $(1 + ds)$, $d\theta$ e $d\mathbf{t}$ é necessário resolver a Equação 6.30 para $d\mathbf{x}$.

$$M(s_i(1 + ds), \theta_i + d\theta)[\mathbf{x}_l + d\mathbf{x}] + \mathbf{t}_i + d\mathbf{t} = \mathbf{x}_i + d\mathbf{x}_i \quad (6.30)$$

As manipulações matemáticas mostradas a seguir ilustram como se chega na Equação 6.31.

$$M(s_i(1 + ds), \theta_i + d\theta)[\mathbf{x}_l + d\mathbf{x}] = \mathbf{x}_i + d\mathbf{x}_i - (\mathbf{t}_i + d\mathbf{t})$$

Usando $\mathbf{x}_i = M(s_i, \theta_i)[\mathbf{x}_l] + \mathbf{t}_i$, obtém-se:

$$M(s_i(1 + ds), \theta_i + d\theta)[\mathbf{x}_l + d\mathbf{x}] = M(s_i, \theta_i)[\mathbf{x}_l] + d\mathbf{x}_i - d\mathbf{t}$$

Uma vez que $M^{-1}(s, \theta)[\dots] = M(s^{-1}, -\theta)[\dots]$, tem-se:

$$\mathbf{x}_l + d\mathbf{x} = M\left((s_i(1 + ds))^{-1}, -(\theta_i + d\theta)\right) \left[M(s_i, \theta_i)[\mathbf{x}_l] + d\mathbf{x}_i - d\mathbf{t} \right]$$

que resulta em:

$$d\mathbf{x} = M\left((s_i(1 + ds))^{-1}, -(\theta_i + d\theta)\right) \left[M(s_i, \theta_i)[\mathbf{x}_l] + d\mathbf{x}_i - d\mathbf{t} \right] - \mathbf{x}_l \quad (6.31)$$

Em geral, o vetor resultante $d\mathbf{x}$ está no espaço $2n$ -D, porém, como só existem t (menor que $2n$) modos de variação descritos no modelo, só é possível mover a forma em t dimensões descritas pelos primeiros t eixos principais [35]. Portanto, procura-se o vetor que seja o mais similar possível a $d\mathbf{x}$ mas que caia no espaço t -D. Adotando a aproximação por mínimos quadráticos, a solução $d\mathbf{x}'$ é a projeção de $d\mathbf{x}$ no espaço t -D (espaço definido pelos vetores das componentes principais, ou seja, t colunas de $\mathbf{P}$). Representando matematicamente tem-se a Equação 6.32 [42].

$$d\mathbf{x}' = \mathbf{A}d\mathbf{x} \quad (6.32)$$

onde $\mathbf{A}$ é a matriz de projeção obtida por [71]: $\mathbf{A} = \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T$

Uma vez que as colunas de $\mathbf{P}$ são ortogonais e $\mathbf{P}$ não é mais uma matriz quadrada, tem-se $\mathbf{P}^T\mathbf{P} = \mathbf{I}$, e então $d\mathbf{x}' = \mathbf{P}\mathbf{P}^T d\mathbf{x}$.

Portanto, ao invés de $\mathbf{x}_l$ se mover para $\mathbf{x}_l + d\mathbf{x}$, ele irá se mover para $\mathbf{x}_l + d\mathbf{x}'$. Expressando $d\mathbf{x}'$ como $d\mathbf{x}' = \mathbf{P}d\mathbf{b}'$ e multiplicando ambos os lados por $\mathbf{P}^T$, chega-se à Equação 6.33.

$$d\mathbf{b}' = \mathbf{P}^T d\mathbf{x}' \quad (6.33)$$

6.2.5 Aplicação dos parâmetros encontrados para a modificação da pose e para a deformação da forma

Uma vez encontrados todos os parâmetros para a variação da forma pode-se atualizar os parâmetros de pose e deformação da estimativa inicial. Obtém-se a nova estimativa $\mathbf{x}_i^{(1)}$ pela Equação 6.34.

$$\mathbf{x}_i^{(1)} = M(s_i(1 + ds), \theta_i + d\theta) [\mathbf{x}_l + \mathbf{P}d\mathbf{b}'] + \mathbf{t}_i + d\mathbf{t} \quad (6.34)$$

Então, todo o processo se reinicia considerando $\mathbf{x}_i^{(1)}$ como a forma inicial

e calcula-se $\mathbf{x}_i^{(2)}$. O processo é repetido até que as mudanças na forma se tornem insignificantes.

Escrevendo de um modo diferente, os parâmetros são atualizados de acordo com as representações da Equação 6.35.

$$\begin{aligned}
t_{xi} &\longrightarrow t_{xi} + dt_x \\
t_{yi} &\longrightarrow t_{yi} + dt_y \\
s_i &\longrightarrow s_i(1 + ds) \\
\theta_i &\longrightarrow \theta_i + d\theta \\
\mathbf{b}_l &\longrightarrow \mathbf{b}_l + d\mathbf{b}'
\end{aligned} \tag{6.35}$$

Pode ser utilizada uma versão ponderada da modificação dos parâmetros como mostrado na Equação 6.36.

$$\begin{aligned}
t_{xi} &\longrightarrow t_{xi} + w_t dt_x \\
t_{yi} &\longrightarrow t_{yi} + w_t dt_y \\
s_i &\longrightarrow s_i(1 + w_s ds) \\
\theta_i &\longrightarrow \theta_i + w_\theta d\theta \\
\mathbf{b}_l &\longrightarrow \mathbf{b}_l + \mathbf{W}_b d\mathbf{b}'
\end{aligned} \tag{6.36}$$

onde w_t , w_s e w_θ são escalares e $\mathbf{W}_b$ é uma matriz diagonal de pesos. $\mathbf{W}_b$ geralmente é uma matriz identidade, mas alternativamente, pode possuir pesos proporcionais ao desvio padrão dos parâmetros da forma correspondente ao longo do conjunto de treinamento. Deve-se sempre lembrar de limitar o parâmetro $\mathbf{b}$ (como explicado na sub-seção 6.1.7) para que ele origine formas aceitáveis e permitidas.

6.2.6 Efeito da utilização de imagens em multi-resolução

Na sub-seção 6.1.2 foi mostrado como gerar imagens em multi-resolução. Nesta seção será explicado porque a utilização de uma estrutura em multi-

resolução torna o algoritmo mais rápido e evita que o processo de busca da forma em uma imagem fique preso a mínimos locais sem alcançar o resultado esperado.

Durante o estágio de treinamento é construído um modelo estatístico do perfil das intensidades dos pixels ao longo de uma linha normal passando por cada ponto da forma, como explicado na sub-seção 6.1.4. Depois, na etapa de busca pela forma, constrói-se um perfil de busca de comprimento (n_s), como explicado na seção 6.2.2. O parâmetro n_s afeta consideravelmente o processo de busca da forma. A escolha do comprimento do perfil de busca possui dois requerimentos contraditórios; por um lado, o perfil de busca deve ser longo o suficiente para incluir o ponto alvo, ou seja, o ponto para onde se deseja que o ponto da forma seja movido. Por outro lado, é interessante que o comprimento do perfil de busca seja pequeno por dois motivos principais: primeiro, para reduzir o tempo de processamento, e segundo, se o comprimento do perfil de busca é muito grande e o ponto alvo está próximo da posição atual do ponto da forma, então, existirá uma alta probabilidade do ponto ser movido para regiões distantes e perder o alvo.

Esses fatores sugerem uma aproximação por multi-resolução, em que, o processo de busca é executado, primeiramente, na imagem do topo da pirâmide de imagens mostrada na sub-seção 6.1.2, que é a imagem mais grosseira (com o menor número de linhas e colunas). Depois, a busca é reiniciada na imagem do nível abaixo usando as informações de saída do nível anterior, e assim por diante, até que a base da pirâmide (imagem original) seja processada.

Para que seja possível executar o processo de busca em todos os níveis de resolução, é necessário ter informações a respeito do perfil de intensidade dos pontos da forma em cada um desses níveis. Para isso, durante o treinamento, deve-se obter a média da derivada normalizada do perfil para cada ponto da

forma, em cada nível de resolução que é utilizado.

Independente de qual seja o nível de resolução processado, é utilizada a mesma quantidade de pixels para gerar o perfil de cada ponto da forma. Uma vez que os pixels do nível L possuem 2^L vezes o tamanho dos pixels da imagem original, os modelos dos níveis mais grosseiros representam uma área maior da imagem, como pode ser visto na Figura 6.12. Similarmente, durante a etapa de busca, a procura do ponto alvo é executada sempre ao longo de um comprimento n_s em torno de cada ponto da forma. Os níveis de resolução mais grosseiros permitirão movimentos maiores, fazendo com que o modelo convirja para uma boa solução. Em níveis de resolução mais refinados, esta solução só precisa ser modificada em pequenas proporções.

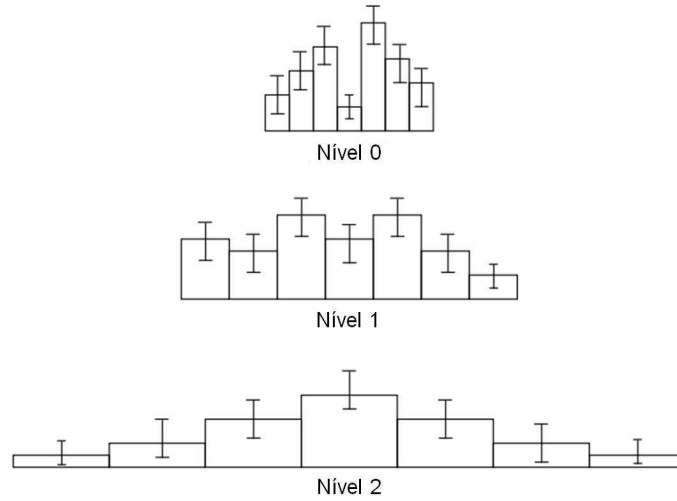


Figura 6.12: *Efeito de se utilizar a mesma quantidade de pixels para gerar o perfil de cada ponto da forma [12].*

Algum critério deve ser estabelecido para que o nível de resolução da busca vá mudando ao longo da pirâmide. Cootes [12] sugeriu mover para o nível abaixo quando uma certa porcentagem dos pontos da forma não mudarem consideravelmente, por exemplo, quando 95% dos pontos da forma moverem apenas em torno de 50% do ponto central do perfil de busca. Este

foi o critério utilizado neste trabalho. Também é interessante estabelecer um número máximo de iterações permitidas, com o objetivo de evitar que o processo fique parado em algum nível alto da pirâmide.

6.3 Pré-Processamento das Imagens de Íris

O ASM busca na imagem por um objeto que tenha uma forma semelhante à do modelo definido. A eficiência do processo aumenta quando esse objeto está bem destacado do resto da imagem, isto é, quando os valores de intensidade dos pixels que formam o objeto são bem diferentes dos pixels próximos às extremidades do objeto.

A grande maioria das imagens de íris utilizadas neste trabalho (provenientes do banco de imagens UBIRIS [63]) apresentam relativamente pouco contraste entre a região da pupila e a região da íris e, muitas vezes, parte da região da íris é sobreposta pelas pálpebras e/ou cílios. Quando isso acontece, o contorno da borda externa da íris fica disforme e despadronizado enquanto os níveis de intensidade se misturam dificultando a identificação do contorno da região pertencente à íris pelo ASM.

Este é um problema típico da utilização de ASM para segmentar a íris, assim, na tentativa de minimizá-lo, foi elaborado um algoritmo de pré-processamento ao qual as imagens foram submetidas antes de serem utilizadas no algoritmo do ASM.

O pré-processamento consiste em aplicar técnicas de realce com a finalidade de diferenciar melhor as regiões da esclera (e pálpebras), da íris e da pupila, além de minimizar as interferências dos cílios. Foi definida a seguinte sequência de operações:

1. Minimização dos efeitos do reflexo especular na pupila;
2. Aplicação de um filtro de mediana 3 x 3; e

```

1  Isaida = I;
2
3  % Corta imagem nas proximidades da pupila:
4  Icortada = Imagem original (I) indo da linha
5             (nTotalLinhas/2-nTotalLinhas/4)
6             até a linha (nTotalLinhas/2+nTotalLinhas/4) e
7             da coluna (nTotalColunas/2-nTotalColunas/5)
8             até a coluna (nTotalColunas/2+nTotalColunas/5);
9
10 Para todos os pixels de Icortada faça
11     Se Icortada(x,y) > 160
12         x_Tamanho_Original = x+(nTotalLinhas/2-nTotalLinhas/4-1);
13         y_Tamanho_Original = y+(nTotalcolunas/2-nTotalcolunas/5-1);
14         Isaida(x_Tamanho_Original,y_Tamanho_Original)=30;
15     Fim Se
16 Fim Para

```

Código 6.3.1: *Rotina para minimizar os efeitos do reflexo especular na pupila.*

3. Aplicação de um alargamento de contraste adaptado em que a transformação de intensidade implementada depende das características da imagem de entrada.

O primeiro passo do algoritmo se fez necessário uma vez que, em todas as imagens do banco UBIRIS, existe a presença de uma região de reflexo especular localizada, na maioria das vezes, sobre a pupila e muitas vezes corrompendo a borda que separa a pupila da íris. Para minimizar o efeito dos reflexos, a idéia foi tentar identificar, nas proximidades da pupila, os pixels com intensidades relativamente altas e substituir estas intensidades por um nível constante e mais escuro que se acredita ser um nível de intensidade típico da região da pupila. O pseudocódigo 6.3.1 mostra como essa idéia foi implementada.

Como pode ser visto no pseudocódigo, inicialmente, a imagem original é cortada com o objetivo de analisar apenas as áreas próximas ao centro da imagem, ou seja, próximas à pupila, onde a ocorrência de interferência por reflexo é mais provável e mais prejudicial. Nesta imagem cortada, os

pixels com intensidade maior que 160 (níveis mais claros) têm sua intensidade modificada para 30 (um nível escuro comum na região da pupila). A Figura 6.13 mostra o resultado do primeiro passo de pré-processamento em algumas imagens.

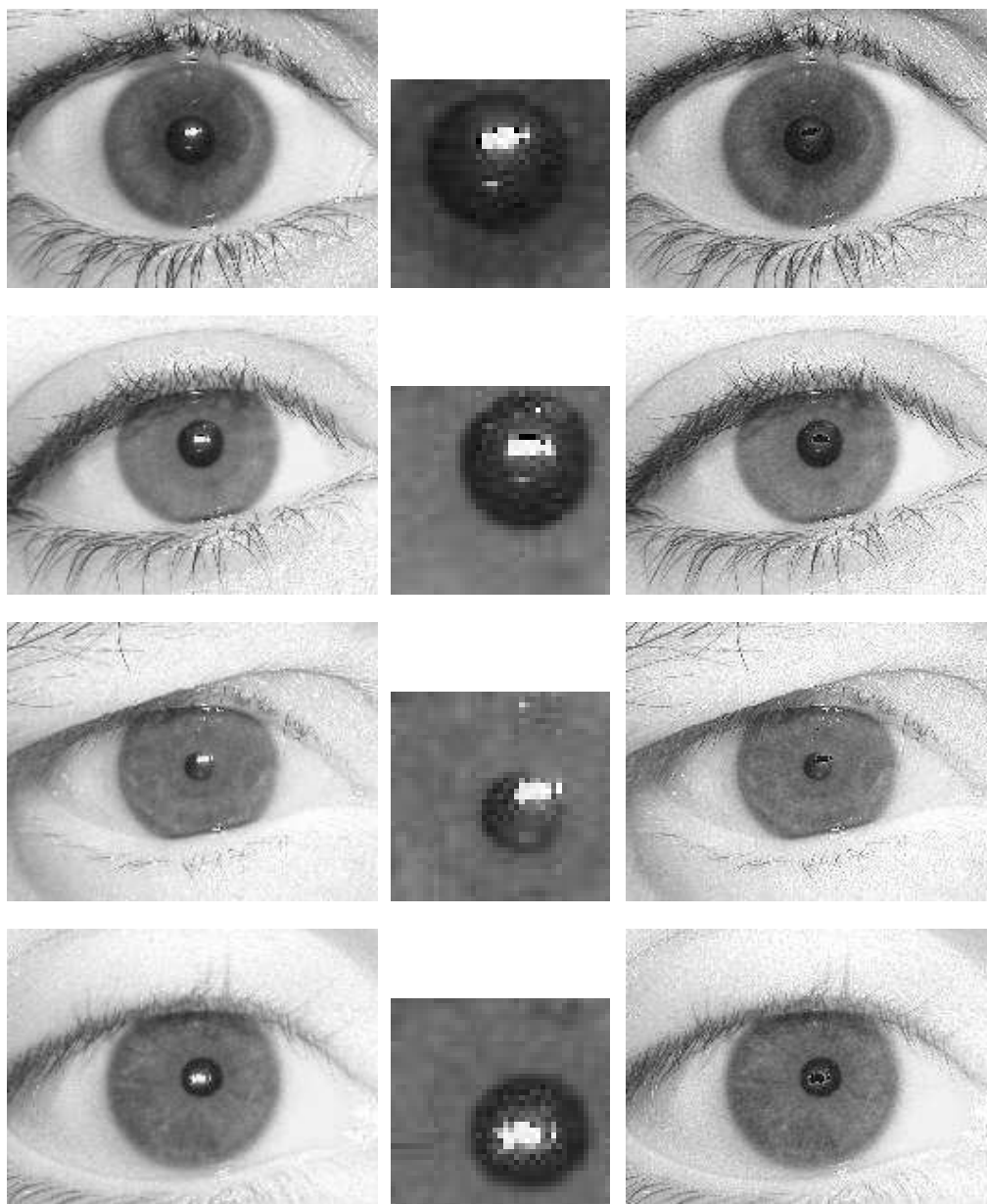


Figura 6.13: *Lado esquerdo: Imagem original; Meio: Imagem cortada; Lado direito: Resultado do primeiro passo de pré-processamento.*

A imagem de saída do primeiro passo de pré-processamento é submetida ao segundo passo, em que ela passa por um filtro de mediana 3 x 3. Para realizar a filtragem de mediana faz-se uma varredura na imagem substituindo o valor de cada pixel pela mediana dos valores de intensidade na vizinhança 3 x 3 desse pixel (o valor original do pixel processado é incluído no cálculo da mediana). A mediana, ξ , de um conjunto de valores é tal que metade dos valores do conjunto é menor ou igual a ξ e a outra metade é maior ou igual a ξ .

A principal função dos filtros de mediana é forçar pontos com níveis de intensidade distintos para serem mais semelhantes aos seus vizinhos [34]. Este processo traz dois benefícios principais: elimina os cílios mais finos da imagem e torna os níveis de intensidade das diferentes regiões da imagem mais homogêneos. A Figura 6.14 mostra o resultado da aplicação do filtro de mediana nas imagens do lado direito da Figura 6.13.

O objetivo principal do pré-processamento é tentar destacar a região da íris do resto da imagem e para isso é importante diferenciar o máximo possível os níveis de intensidade de três conjuntos de pixels: os pixels que pertencem à pupila e aos cílios (que são mais escuros), os pixels da região da íris (que possuem níveis de intensidade intermediários) e os pixels da esclera e das pálpebras (que são mais claros). A alternativa encontrada para fazer isso foi aplicar a transformação de intensidade mostrada no gráfico da Figura 6.15 que relaciona as intensidades da imagem de entrada (no eixo horizontal, r) com as da imagem de saída (no eixo vertical, s). Cada nível de intensidade que aparecer na imagem de entrada tem um nível de intensidade correspondente (mostrado no gráfico) que deverá substituí-lo na imagem de saída.

A imagem de saída é gerada da seguinte forma:

- Para os pixels da imagem de entrada com nível de intensidade entre 0

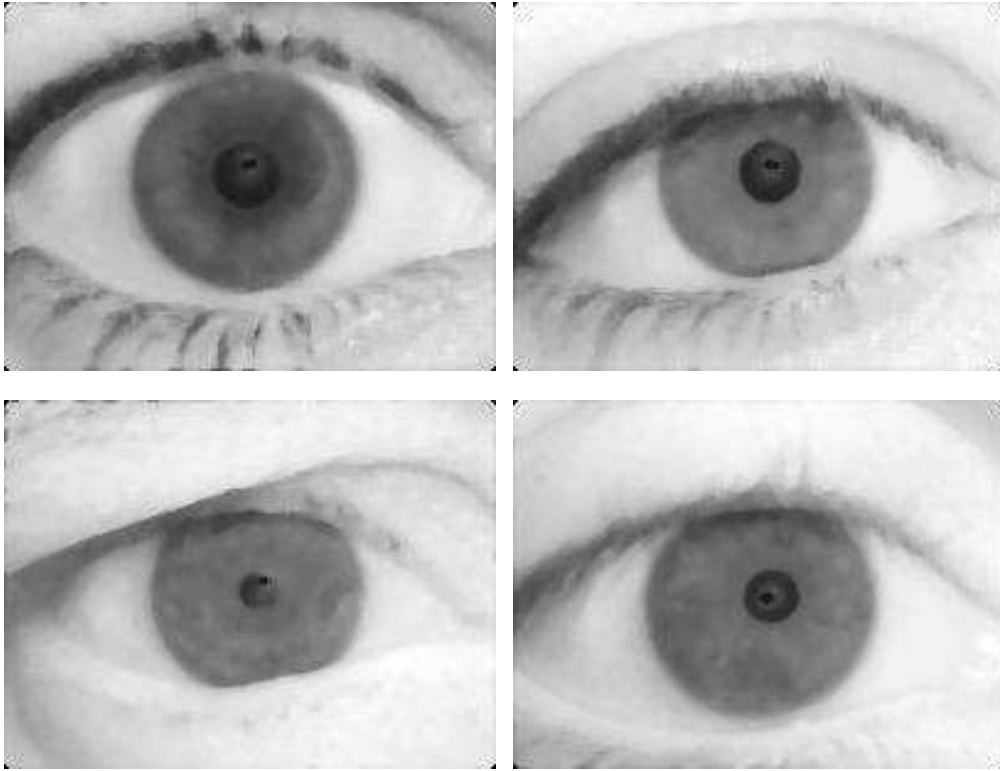


Figura 6.14: *Resultado do segundo passo de pré-processamento.*

e r_1 , aplique a transformação da reta 1.

- Para os pixels da imagem de entrada com nível de intensidade entre r_1 e r_2 , aplique a transformação da reta 2.
- Para os pixels da imagem de entrada com nível de intensidade entre r_2 e 255, aplique a transformação da reta 3.

As equações das retas 1, 2 e 3 são mostradas na Equação 6.37

$$\begin{aligned}
 \text{Reta 1} &\rightarrow s = \frac{s_{11}}{r_1}r \\
 \text{Reta 2} &\rightarrow s = \frac{(s_{21}-s_{12})}{(r_2-r_1)}(r - r_1) + s_{12} \\
 \text{Reta 3} &\rightarrow s = \frac{(255-s_{22})}{(255-r_2)}(r - r_2) + s_{22}
 \end{aligned} \tag{6.37}$$

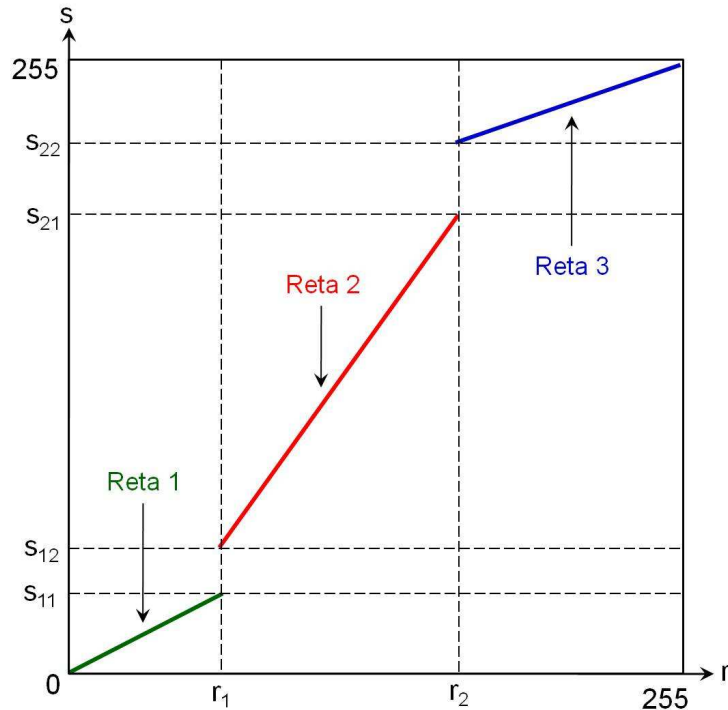


Figura 6.15: *Função de transformação de intensidade implementada para o alargamento de contraste.*

O interessante da transformação de alargamento de contraste implementada é que os parâmetros r_1 , r_2 , s_{11} , s_{12} , s_{21} e s_{22} não foram considerados constantes para todas as imagens, eles foram definidos em função das características estatísticas de cada imagem. A seguir descreve-se o processo de definição desses parâmetros:

a) **Definição de r_1**

Foi constatado empiricamente que, na maioria das imagens utilizadas, os pixels pertencentes à pupila e aos cílios (pixels mais escuros) correspondem a aproximadamente 3% do total de pixels da imagem. Assim, r_1 foi definido como sendo o nível de intensidade tal que 3% do total de pixels da imagem possuem intensidade menor do que ele.

b) **Definição de r_2**

A escolha ideal para r_2 seria o nível de intensidade que separa os pixels mais claros da imagem (pertencentes às pálpebras e à esclera) do restante da imagem.

O método de limiarização de Otsu [57] escolhe um limiar global que minimiza a variância das duas classes formadas no processo de binarização. Experimentando binarizar as imagens de íris observou-se que o método de Otsu escolhe justamente um limiar que separa relativamente bem as regiões da imagem pertencentes às pálpebras e à esclera das demais partes da imagem. Isto pode ser comprovado observando as imagens binarizadas da Figura 6.16. Os detalhes do método de Otsu são apresentados no Apêndice B.

Assim, r_2 foi definido como sendo o nível de intensidade calculado como limiar global pelo método de Otsu.

c) **Definição de s_{11} , s_{12} , s_{21} e s_{22}**

Os parâmetros s_{11} , s_{12} , s_{21} e s_{22} são definidos em função dos valores de r_1 e r_2 como mostrado na Equação 6.38

$$\begin{aligned} s_{11} &= 0,7r_1 \\ s_{12} &= 0,9r_1 \\ s_{21} &= 1,25r_2 \\ s_{22} &= 1,3r_2 \end{aligned} \tag{6.38}$$

Com essas equações observa-se que s_{11} é 30% menor que r_1 , s_{12} é 10% menor que r_1 , s_{21} é 25% maior que r_2 e s_{22} é 30% maior que r_2 .

Definindo os parâmetros dessa forma, os níveis de intensidade entre 0 e r_1 são ligeiramente escurecidos, os níveis de intensidade entre r_1 e r_2 são

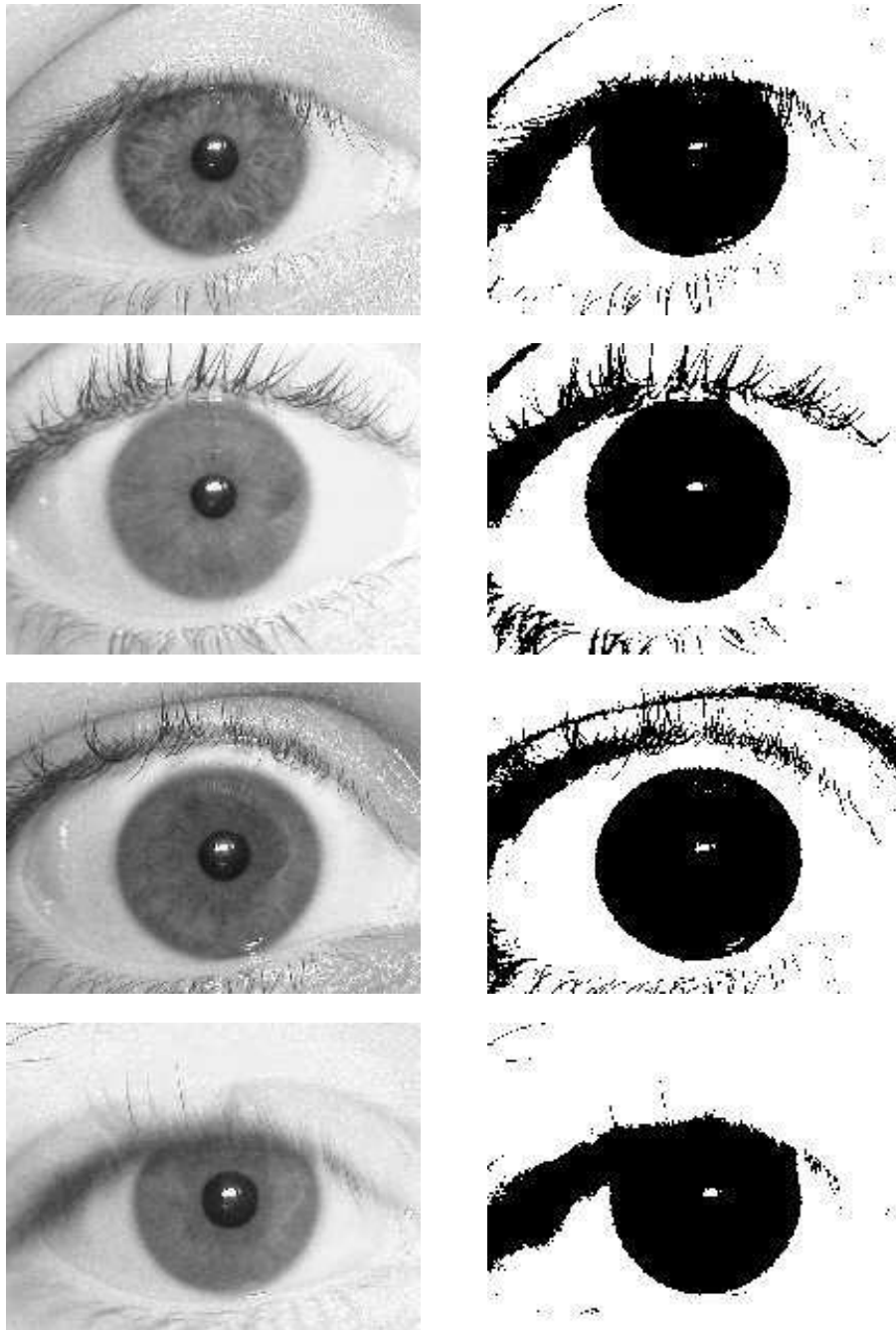


Figura 6.16: *Lado esquerdo: Imagem original; Lado direito: Imagem binarizada pelo método de Otsu.*

ligeiramente clareados e os níveis de intensidade entre r_2 e 255 ficam bem mais claros.

A Figura 6.17 mostra alguns resultados obtidos com esse último passo de pré-processamento. O alargamento de contraste projetado foi aplicado à imagens resultantes da filtragem de mediana. Observa-se que, em geral, o pré-processamento melhorou o contraste da região da íris em relação ao restante da imagem o que favorecerá a atuação do ASM.

6.4 Resultados da Utilização de ASM para Segmentar a Região da Íris

Após o entendimento da teoria utilizada na implementação do Active Shape Models (ASM), foram realizados testes da utilização de ASM para segmentar a região da íris em imagens de olho do banco de imagens UBIRIS (Sessão 1) [63].

É importante ressaltar que, neste trabalho, a intenção foi realizar experimentos preliminares para avaliar o comportamento do algoritmo e identificar os principais obstáculos e desafios da utilização do ASM para segmentar a íris.

O algoritmo do ASM foi aplicado a todas as 1201 imagens do banco de imagens citado. Todas essas imagens passaram pelo pré-processamento descrito na seção 6.3, tanto na etapa de treinamento quanto na etapa de busca pela forma.

Como foi mencionado ao longo das seções 6.1 e 6.2, diversos parâmetros devem ser definidos e ajustados para a utilização do ASM. A definição de todos os parâmetros da etapa de treinamento está resumida na Tabela 6.1 e a da etapa de busca da forma, na Tabela 6.2.

O efeito da variação de todos os parâmetros envolvidos no processo foi detalhado ao longo das seções 6.1 e 6.2.

Após aplicar o ASM em todas as imagens, foi realizada uma inspeção

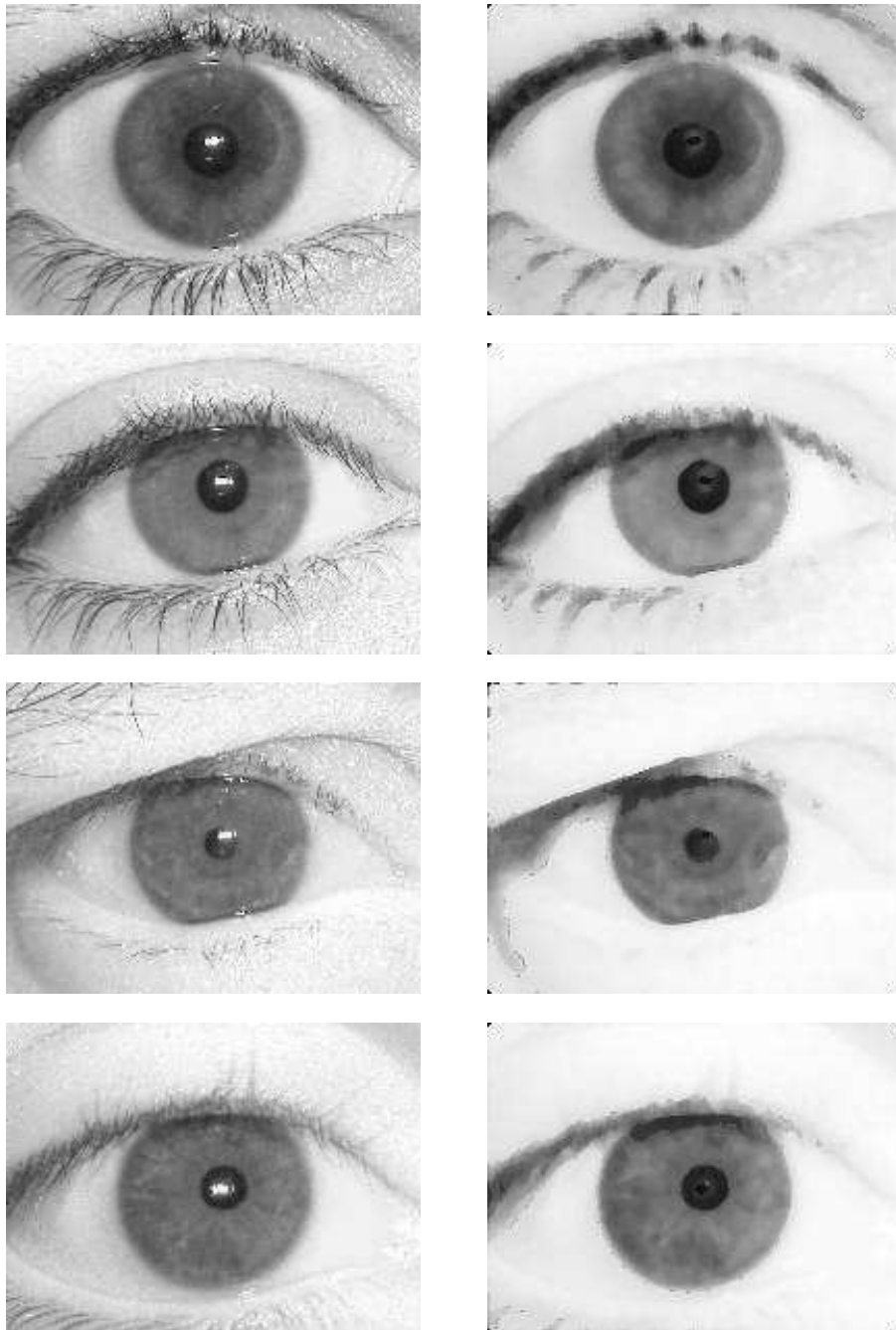


Figura 6.17: *Lado esquerdo: Imagem original; Lado direito: Resultado final do algoritmo de pré-processamento.*

visual nos resultados obtidos. Ao invés de classificar os resultados simplesmente em segmentação correta e segmentação com erros, os erros encontrados

Tabela 6.1: *Definição dos parâmetros da etapa de treinamento.*

Parâmetro	Descrição	Valor assumido
N	Número de imagens do conjunto de treinamento	100
n	Número de pontos da forma	80
$Nivel$	Máximo nível de multi-resolução utilizado	2
n_{Acima}	Número de pontos acima do ponto da forma utilizados para gerar o perfil desse ponto	5
n_{Abaixo}	Número de pontos abaixo do ponto da forma utilizados para gerar o perfil desse ponto	3
f_v	Porcentagem da variância do conjunto de treinamento que se deseja preservar ao utilizar a representação por PCA	0,5
m	Parâmetro utilizado para limitar a variação do vetor $\mathbf{b}$ a, no máximo, m vezes o desvio padrão dos pontos	3

Tabela 6.2: *Definição dos parâmetros da etapa de busca.*

Parâmetro	Descrição	Valor assumido
n_s	Número de pontos do perfil de busca	$2n_{Acima} + 2n_{Abaixo} + 1 = 17$
$maxIter$	Número máximo de iterações que são executadas para buscar a forma na imagem	10

foram categorizados, com o objetivo de identificar as principais dificuldades impostas ao algoritmo do ASM pelas imagens utilizadas, o que pode ser útil no direcionamento de futuros trabalhos que poderão ser desenvolvidos.

A estatística dos resultados é apresentada na Tabela 6.3.

Pode-se verificar que 50,54% das imagens tiveram uma segmentação que

Tabela 6.3: *Resumo dos resultados da aplicação do algoritmo do ASM.*

	Quantidade de imagens	Porcentagem do total
Segmentação correta	607	50,54%
Erro na segmentação da pálpebra superior	325	27,06%
Erro na segmentação da pupila	119	9,91%
Erro na segmentação da pálpebra superior e da pupila	35	2,91%
Erro na segmentação da borda externa da íris	75	6,24%
Erros graves	40	3,33%
TOTAL	1201	100%

pode ser considerada sem defeito. A Figura 6.18 ilustra alguns desses resultados mostrando, do lado esquerdo, a forma inicial (definida na etapa de treinamento) e, do lado direito, a forma final obtida pelo ASM no processo de busca.

Os erros foram classificados nas seguintes categorias:

- **Erro na segmentação da pálpebra superior**

Resultados nos quais a segmentação não foi eficiente apenas nas áreas da região da íris sobrepostas pela pálpebra superior. Nesses resultados, a região da íris encontrada possui áreas de interferência da pálpebra superior.

- **Erro na segmentação da pupila**

Resultados nos quais a segmentação não foi eficiente apenas nos limites da região da íris com a pupila.

- **Erro na segmentação da pálpebra superior e da pupila**

Resultados nos quais a segmentação não foi eficiente tanto nas áreas da região da íris sobrepostas pela pálpebra superior quanto nos seus limites com a pupila.

- **Erro na segmentação da borda externa da íris**

Resultados nos quais a segmentação não teve sucesso na detecção dos limites da borda externa da íris, porém, a região da íris encontrada não sofre interferências de áreas não pertencentes à íris.

- **Erros graves**

Resultados que não pertencem a nenhuma das categorias anteriores ou que acumulam os erros representados em pelo menos duas dessas categorias.

Observando a Tabela 6.3, verifica-se que o principal problema que ocorre é o erro na detecção dos limites da região da íris nas áreas sobrepostas pela pálpebra superior. Este resultado pode ser justificado, principalmente, pelo fato de que quando a pálpebra sobrepõe a íris, geralmente, ocorre a interferência dos cílios que muitas vezes deformam a borda de separação entre a pálpebra e a íris. Além disso, o efeito aleatório provocado pela presença dos cílios e também de uma sombra que às vezes a pálpebra produz na região da íris, dificulta a representação de todas as possíveis variações da forma no conjunto de treinamento. A Figura 6.19 ilustra alguns exemplos de erro na segmentação da pálpebra superior.

A ocorrência de erros na segmentação da pupila se deve, principalmente, à interferência dos reflexos especulares que algumas vezes deformam a borda de separação entre a região da íris e a pupila e, outras vezes, confundem o algoritmo do ASM. Também ocorre casos em que o erro é provocado pelo fato da íris ser muito escura e existir pouca diferença de contraste entre ela e

a pupila. Alguns exemplos de erro na segmentação da pupila são ilustrados na Figura 6.20.

Em 2,91% dos resultados ocorre, ao mesmo tempo, o erro na segmentação da pálpebra superior e na segmentação da pupila. Exemplos desses resultados são ilustrados na Figura 6.21.

Os erros na segmentação da borda externa da íris ocorrem quando o contraste entre a região da íris e a esclera não é alto o suficiente para ser percebido e interpretado pelo algoritmo do ASM ou quando as características da imagem não são contempladas no conjunto de treinamento. A Figura 6.22 ilustra alguns exemplos de erro na segmentação da borda externa da íris.

Finalmente, alguns exemplos de resultados considerados erros graves são ilustrados da Figura 6.23.

6.5 Avaliação dos Resultados Obtidos

Conforme apresentado na Tabela 6.3 o ASM teve uma eficiência de 50,54% na segmentação da região da íris. Com esta taxa de acerto, o ASM ainda não pode ser considerado uma técnica preferencial para ser utilizada para compor um sistema de reconhecimento de íris. Porém, como se tratou de uma primeira tentativa, este resultado pode ser considerado encorajador, indicando que as informações e as experiências adquiridas com esta tese podem servir de inspiração para trabalhos futuros que visem aperfeiçoar a utilização da técnica no sentido de melhorar seus resultados.

Como a principal dificuldade encontrada pelo ASM foi a segmentação da região da íris nas áreas sobrepostas pela pálpebra superior, uma alternativa que poderia tornar sua utilização mais viável seria associar um algoritmo eficiente para fazer a detecção e a exclusão da pálpebra superior.

É importante lembrar que, neste trabalho, todo o contorno da região da

íris foi representado por uma única curva. Existe a possibilidade de definir o modelo de uma curva (ou forma) para a borda externa da íris e outro para a borda da pupila e aplicar o ASM para encontrar esses modelos na imagem independentemente. Esta alternativa pode tornar o algoritmo mais robusto na segmentação dos limites da borda externa da íris com a pálpebra e também pode amenizar os efeitos do segundo tipo de erro mais incidente que foi o erro na segmentação da pupila.

Outro recurso para tentar melhorar a eficiência do ASM é fazer modificações nos diversos parâmetros variáveis do algoritmo. Apesar de, neste trabalho, já ter sido realizado um esforço no sentido de obter um bom ajuste dos parâmetros, apenas uma pequena parcela das combinações de valores de parâmetros foram avaliadas. Portanto, seria interessante testar novas possibilidades.

6.6 Conclusões do Capítulo

Este capítulo teve como objetivo apresentar a técnica do Active Shape Model (ASM) e avaliar a sua utilização para segmentar a região da íris.

O ASM é uma técnica que modela uma forma flexível para encontrar um objeto conhecido em uma determinada imagem. O processo do ASM consiste em uma etapa de treinamento do modelo e, posteriormente, uma etapa de busca pela forma em uma imagem. Nas seções 6.1 e 6.2, todos os detalhes da implementação dos algoritmos dessas duas etapas de processamento foram descritos.

Na seção 6.3 foi apresentado o algoritmo de pré-processamento que foi aplicado às imagens de íris com o objetivo de tornar-las mais adequadas à atuação do ASM.

Finalmente, na seção 6.4 foram apresentados os resultados experimentais

da aplicação do algoritmo do ASM para segmentar a região da íris e na seção 6.5 foi realizada uma avaliação crítica desses resultados.

A taxa de acerto do ASM foi de 50,54%, o que sugere que, com os parâmetros que foram utilizados, o algoritmo do ASM ainda não está adequado para ser utilizado em situações práticas. Porém, a análise dos erros obtidos possibilitou sugerir alternativas para lidar com as dificuldades do processo, encorajando o desenvolvimento de diversos trabalhos futuros.

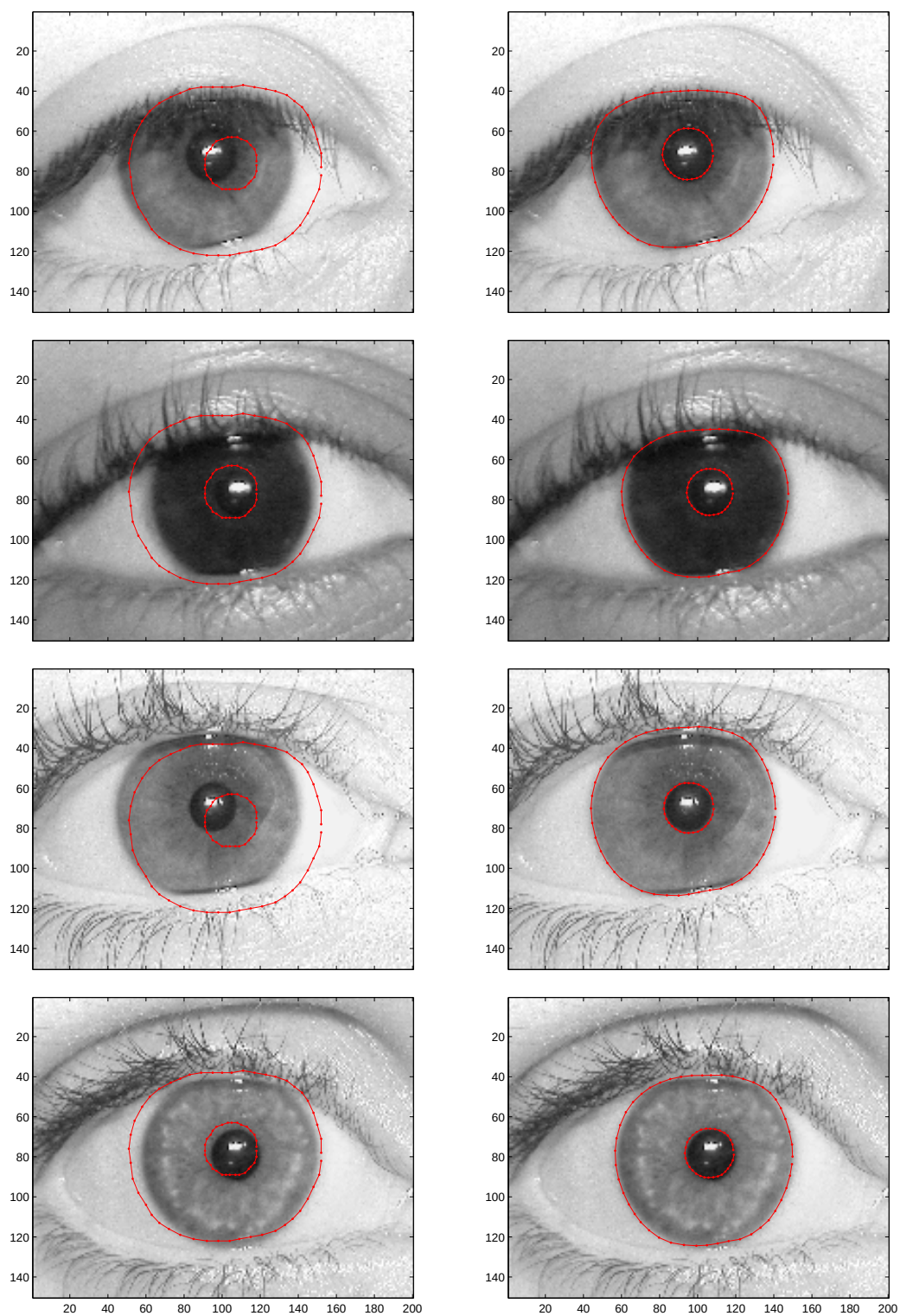


Figura 6.18: Exemplos de resultados de segmentação correta. Lado esquerdo: forma inicial; Lado direito: forma final obtida pelo ASM.

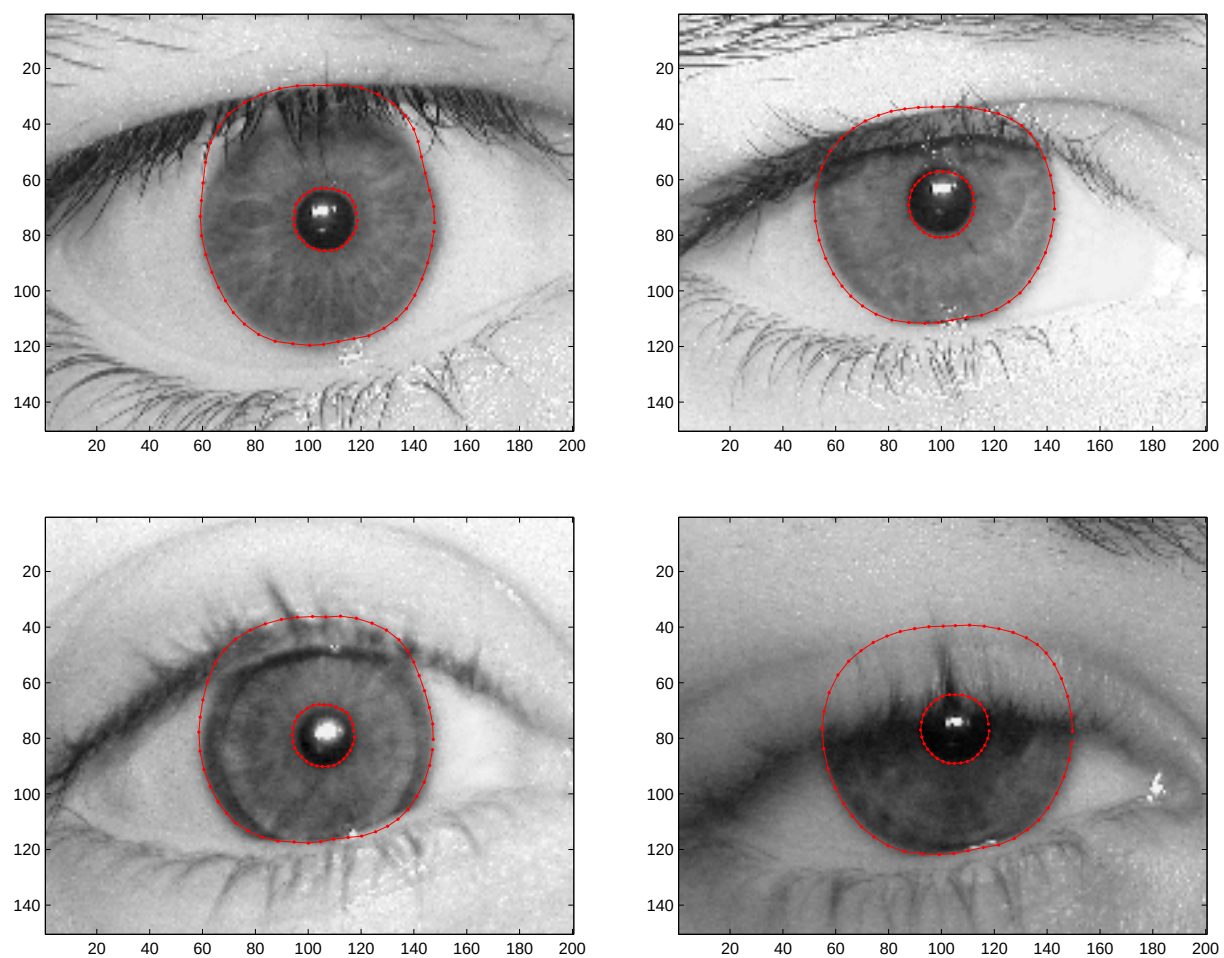


Figura 6.19: *Exemplos de resultados de erro na segmentação da pálpebra superior.*

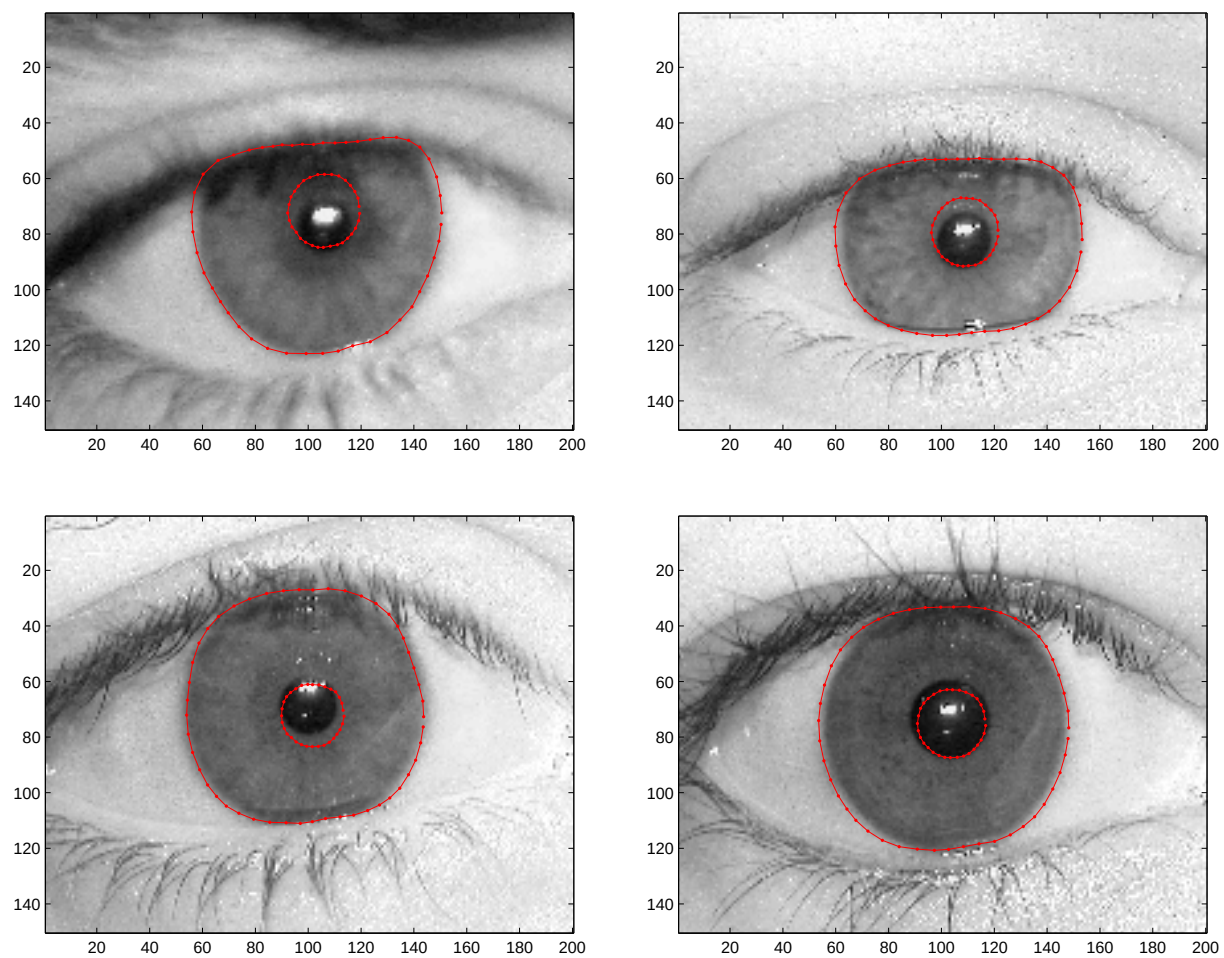


Figura 6.20: *Exemplos de resultados de erro na segmentação da pupila.*

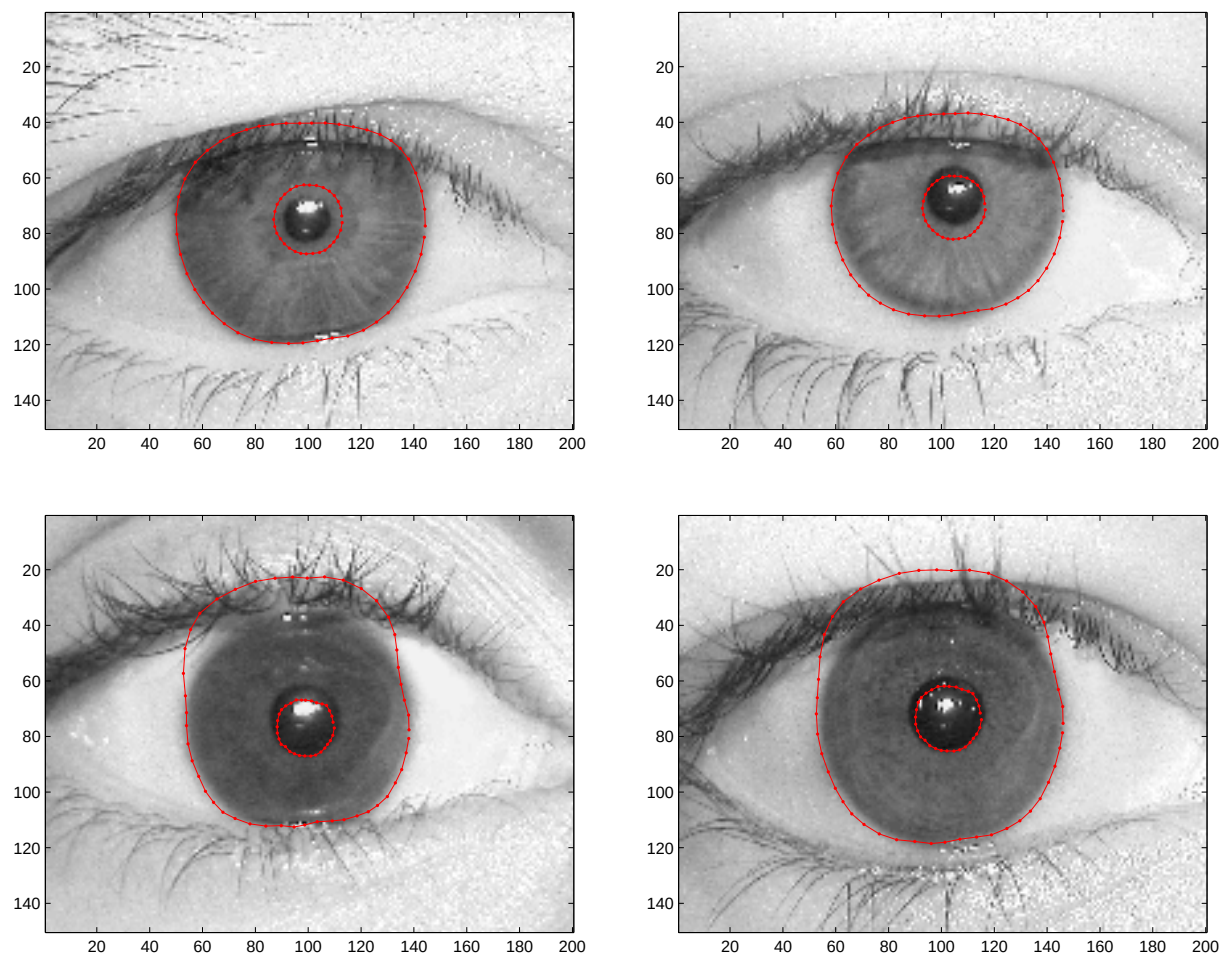


Figura 6.21: *Exemplos de resultados de erro na segmentação da pálpebra superior e da pupila.*

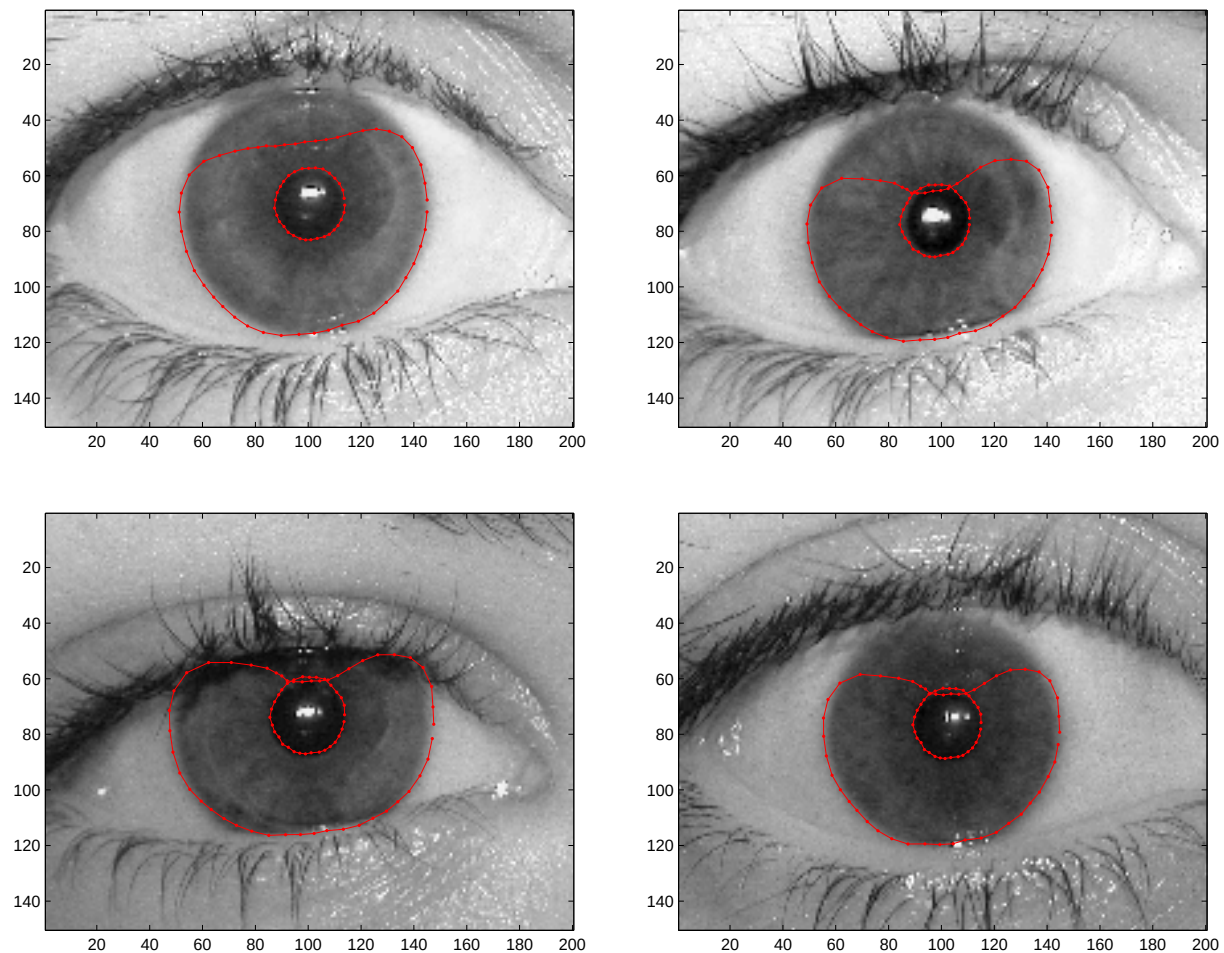


Figura 6.22: *Exemplos de resultados de erro na segmentação da borda externa da íris.*

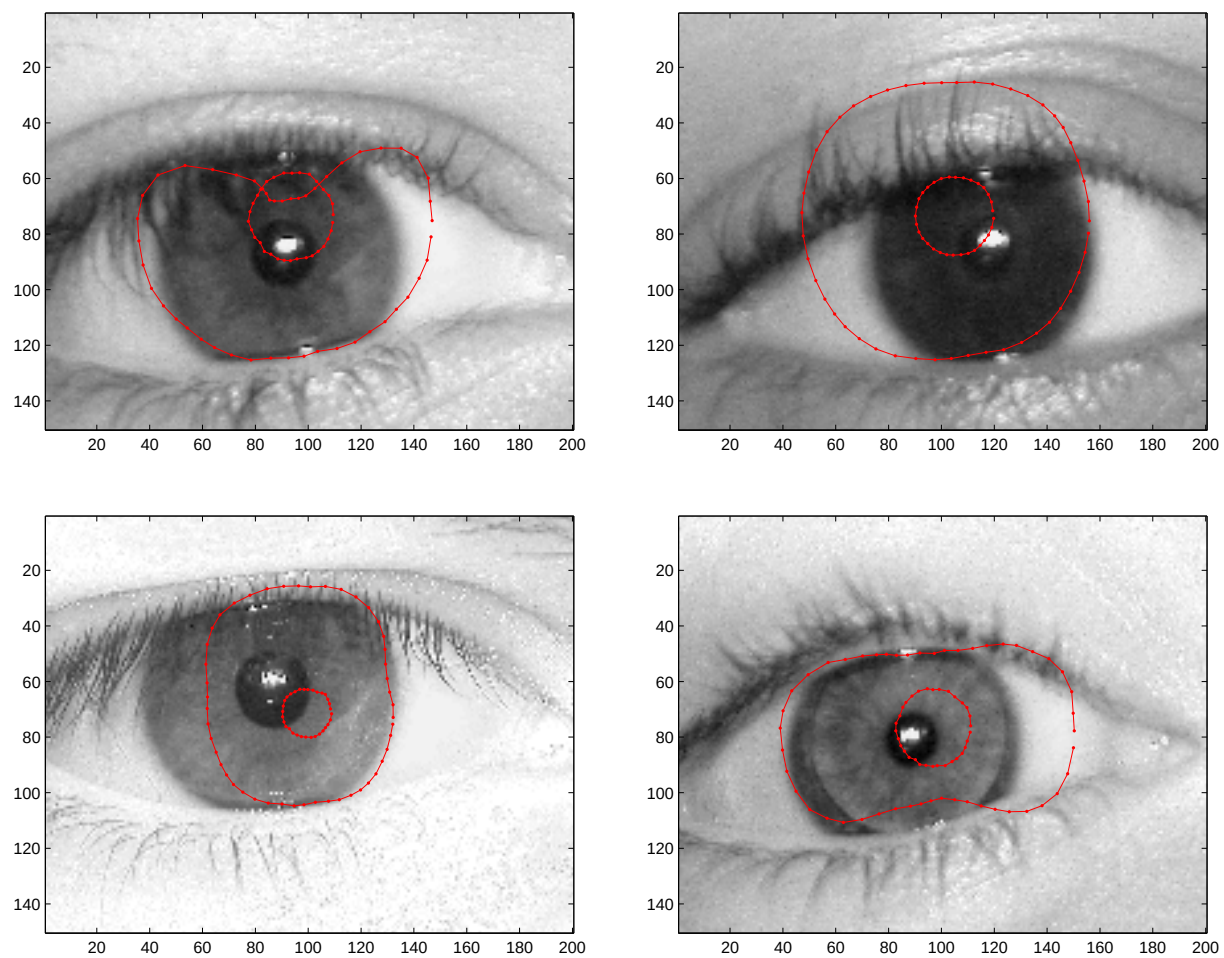


Figura 6.23: *Exemplos de resultados de erros graves.*

Capítulo 7

CONCLUSÕES, CONTRIBUIÇÕES E SUGESTÕES DE TRABALHOS FUTUROS

7.1 Resumo das Implementações e dos Resultados

Na primeira parte deste trabalho, foram descritas as implementações de todas as etapas de processamento de um sistema de reconhecimento de íris utilizando métodos tradicionais. Essas etapas são: localização, normalização, extração e codificação de características e comparação.

Para realizar a localização da região da íris foram utilizadas implementações de dois métodos diferentes, largamente conhecidos e empregados na literatura e que aproximam a região da íris a uma região anelar limitada por dois círculos. Esses métodos são Transformada Circular de Hough (TCH) e Operador Integro-Diferencial que foram descritos na seção 3.1. Nesta mesma seção, também foram descritos o método utilizado para detecção da interferência dos cílios que é baseado no estabelecimento de um valor de limiar e, ainda, o método empregado para detectar as pálpebras que é baseado na

utilização da Transformada Linear de Hough e permite a exclusão das áreas corrompidas por elas.

Na seção 3.2 foi descrito o método utilizado na etapa de normalização cujo objetivo é eliminar as inconsistências dimensionais das imagens utilizadas. Foi implementado o método mais tradicionalmente utilizado na literatura que modela a região da íris como se essa fosse um anel de borracha e, utilizando-se coordenadas polares, gera-se uma representação retangular da íris com dimensões constantes. Esse método foi proposto por John Daugman e envolve uma amostragem uniforme, na região da íris, dos pixels que serão processados.

Na próxima etapa de processamento, as características da íris são extraídas da representação retangular resultante da etapa de normalização e, posteriormente, codificadas para gerar um *template* binário. Para realizar a extração dessas características, a imagem normalizada é convoluída com um filtro Log-Gabor 1D cujos fundamentos foram descritos na seção 3.3. Em seguida, é realizada uma quantização da fase do sinal de saída do filtro e é gerado o *template* binário.

Para realizar a comparação de dois *templates*, que é a última etapa de processamento e serve para definir se os *templates* são provenientes da mesma íris ou não, foi utilizada como métrica a Distância de Hamming (dH). O valor de dH fornece uma medida da quantidade de bits que não são correspondentes entre os dois *templates*. É definido um valor de dH limiar de forma que dois *templates* só são considerados provenientes da mesma íris se o valor de dH gerado pela comparação entre eles for menor do que esse valor de limiar. Na etapa de comparação é tratado o problema de inconsistências rotacionais das imagens utilizadas realizando o deslocamento horizontal do *template* (que representa um deslocamento na direção angular da íris). Assim, são realizados um determinado número de deslocamentos para a direita e para a esquerda, calculando o valor de dH para cada caso, de forma que, no

final, somente o menor valor de dH obtido é considerado. Os detalhes dessa implementação foram apresentados na seção 3.4.

O sistema de reconhecimento de íris implementado com as técnicas tradicionais foi testado utilizando-se dois bancos de imagens de íris. Os parâmetros do sistema foram ajustados e otimizados para cada banco de imagens e os detalhes foram mostrados na seção 3.6. Para o banco de imagens UBIRIS o menor erro de reconhecimento obtido foi de 0,07% que equivale a uma taxa de acerto de 99,93%. Para isto foi utilizada uma representação retangular da íris com resolução 14 x 150 pixels que gera um *template* de 4200 bits. Para o banco de imagens MMU o menor erro de reconhecimento obtido foi de 0,91% que equivale a uma taxa de acerto de 99,09%. Para isto foi utilizada uma representação retangular da íris com resolução 8 x 370 pixels que gera um *template* de 13320 bits.

No capítulo 4 foi apresentado o método proposto para localizar a região da íris utilizando Algoritmos Meméticos. Todos os detalhes da sua implementação e do ajuste de parâmetros foram descritos e os resultados obtidos foram satisfatórios. Dentro de uma classificação existente, o método proposto pode ser considerado um método de localização da íris baseado em fronteira. Assim, ele foi comparado ao método clássico de localização baseado em fronteira sugerido por Wildes que emprega a Transformada Circular de Hough. O método proposto localizou a região da íris em 97,08% das imagens do banco de imagens UBIRIS enquanto a eficiência do método de Wildes foi de 93,17% empregando-se as mesmas imagens. Outro benefício obtido foi o tempo de processamento. O tempo de processamento do método proposto foi cerca de cinco vezes menor do que o do método de Wildes para segmentar a íris.

No capítulo 5 foi apresentado o método de pré-processamento proposto para realizar uma amostragem não-uniforme de pixels na região da íris. A busca pelos pixels que melhor representam as características da íris foi reali-

zada utilizando-se Algoritmos Genéticos. A eficiência do sistema quando esse método é utilizado foi testada usando os bancos de imagens UBIRIS e MMU. Foram realizadas diversas simulações e os resultados obtidos foram apresentados na seção 5.3. Os resultados mostraram que o novo método permitiu aumentar a eficiência do sistema quando é mantido o tamanho do *template* e ainda, permitiu diminuir o tamanho do *template*, mantendo-se a eficiência do sistema em níveis satisfatórios.

No capítulo 6 foi avaliada a utilização da técnica de Active Shape Model (ASM) para segmentar a região da íris. Todos os aspectos da implementação do extenso algoritmo do ASM foram descritos com detalhes com o objetivo de deixar evidente a influência de todos os passos de processamento e de todos os parâmetros envolvidos. Foi desenvolvido também, um algoritmo de pré-processamento para ser aplicado às imagens de íris antes de serem submetidas ao ASM. Esse pré-processamento foi desenvolvido com o objetivo de minimizar algumas dificuldades do algoritmo do ASM. Foram realizados testes experimentais aplicando o ASM à todas as imagens do banco de imagens UBIRIS com o objetivo de segmentar a região da íris. O algoritmo conseguiu segmentar sem defeito 50,54% das imagens e os erros de segmentação obtidos foram separados em categorias de erro. Uma análise dessas categorias de erro permitiu identificar as principais deficiências do algoritmo do ASM e sugerir algumas alternativas para tentar melhorar sua eficiência, encorajando o desenvolvimento de trabalhos futuros.

7.2 Conclusões e Contribuições

Este trabalho foi desenvolvido com a intenção de trazer contribuições para a área de pesquisa de sistemas de reconhecimento biométrico de indivíduos utilizando-se as características da íris. A primeira contribuição que se pode

destacar foi a apresentação do estado da arte que cita os principais trabalhos publicados e as principais técnicas empregadas em sistema de reconhecimento de íris. O estado da arte serve como fonte de informações e referências para aqueles que pretendem se atualizar nessa área.

O método proposto para localizar a íris foi comparado a um método tradicional e apresentou vantagens em relação à eficiência da localização e ao tempo de processamento. Entretanto, a principal contribuição do método é a inovação de se utilizar algoritmos evolutivos nesta etapa de processamento e, ainda, a constatação da eficácia dessa utilização.

Outra contribuição extremamente importante deste trabalho foi o método proposto para amostrar os pixels da região da íris de maneira não uniforme. Os resultados da aplicação desse método mostraram que as principais características da íris que diferenciam os indivíduos podem não estar igualmente distribuídas ao longo de toda a região da íris e a amostragem dos pixels que armazenam mais informações relevantes ao reconhecimento, além de tornar o sistema mais eficiente, permite a construção de templates menores.

O estudo do Active Shape Model foi válido pois tornou evidente que a sua utilização para segmentar a região da íris não é óbvia e necessita de um profundo conhecimento do algoritmo do ASM e das particularidades desta aplicação. Neste sentido, este trabalho forneceu uma contribuição significativa uma vez que foi desenvolvido um modelamento para o problema e foram realizados testes preliminares para identificar as principais dificuldades e desafios dessa aplicação. Outra contribuição importante deste estudo foi o desenvolvimento de um material teórico com o detalhamento completo da implementação do algoritmo do ASM que pode ser utilizado por pesquisadores que têm a intenção de trabalhar com o ASM na área de reconhecimento de íris ou em aplicações de qualquer outra área do conhecimento.

É interessante comentar que, apesar do enorme avanço tecnológico dos

sistemas de reconhecimento de íris atuais, ainda é possível desenvolver novas técnicas que possam contribuir para o aprimoramento da área a nível científico e até mesmo comercial.

7.3 Sugestões de Trabalhos Futuros

Apesar deste trabalho ter atendido a todos os seus objetivos iniciais, o seu desenvolvimento ofereceu condições para que outros trabalhos possam ser sugeridos, como por exemplo:

- Podem ser utilizadas diferentes técnicas em todas as etapas de processamento do sistema de reconhecimento de íris para que seus resultados sejam comparados com os obtidos neste trabalho e em outros trabalhos da literatura científica;
- Todos os algoritmos desenvolvidos neste trabalho foram implementados em MatLab. A implementação destas técnicas em outras linguagens, como C ou C++, pode ser uma alternativa interessante para aumentar a velocidade de processamento dos algoritmos mais onerosos;
- Os métodos propostos neste trabalho podem ser testados em outros bancos de imagens;
- Tanto o método proposto para localizar a íris quanto o método proposto para amostragem não uniforme dos pixels da região da íris foram baseados na implementação de algoritmos evolutivos (genéticos e meméticos) que dependem de ajuste de parâmetros. Novas combinações de parâmetros podem ser testadas; e
- A experiência adquirida neste trabalho com a utilização de Active Shape Models (ASM) para segmentar a região da íris pode inspirar diversos trabalhos como:

- Associar um algoritmo para detecção de pálpebras ao algoritmo do ASM;
- Implementar o ASM com dois contornos, um para a borda externa da íris e outro para a borda da pupila;
- Testar novas combinações dos diversos parâmetros variáveis do algoritmo do ASM;
- Aprimorar o algoritmo de pré-processamento das imagens de íris na tentativa de melhorar o desempenho do ASM; e
- Aplicar o algoritmo do ASM a outros bancos de imagens.

7.4 Publicações

- ICME 2009 - IEEE International Conference on Multimedia and Expo.
 - *Título:* “APPLICATION OF EVOLUTIONARY ALGORITHMS FOR IRIS LOCALIZATION”
- WISP 2009 - 6th IEEE International Symposium on Intelligent Signal Processing.
 - *Título:* “ANALYSING THE PERFORMANCE OF THE ALGORITHMS USED TO LOCALIZE THE IRIS REGION IN EYE IMAGES SUBMITTED TO SEVERELY COMPRESSED IMAGES”
- Applied Artificial Intelligence, Taylor & Francis, 2009, Vol. 23, No. 8, pg. 238-757
 - *Título:* “LOCALIZING THE IRIS THROUGH MEMETIC ALGORITHMS”
- Journal of Communication and Information Systems (JCIS), Brazilian Telecommunications Society and IEEE Communication Society, 2009, Vol. 24, No. 1, pg. 10-17
 - *Título:* “PROCESSING THE SEGMENTATION STAGE OF AN IRIS RECOGNITION SYSTEM THROUGH EVOLUTIONARY ALGORITHM”

Referências Bibliográficas

- [1] *Chinese Academy of Sciences - Institute of Automation, Database of 756 Greyscale Eye Images.* www.sinobiometrics.com.
- [2] *Multimedia University iris image database.* <http://pesona.mmu.edu.my/ccteol/>.
- [3] What are log-gabor filters and why are they good? www.csse.uwa.edu.au/~pk/Research/MatlabFns/PhaseCongruency/Docs/convexpl.html.
- [4] ADLER, F. H. *Physiology of the Eye: Clinical Application.* London: The V. C. Mosby Company, 4^a ed., 1965.
- [5] AL-ZUBI, R. & ABU-AL-NADI, D. I. Automated personal identification system based on human iris analysis. *Pattern Analysis and Applications*, vol. 10, no. 2, 2007. DOI: 10.1007/s10044-006-0058-2.
- [6] ALI, J. M. H. & HASSANIEM, A. E. An iris recognition system to enhance e-security environment based on wavelet theory. *AMO - Advanced Modeling and Optimization Journal*, vol. 5, no. 2, 2003.
- [7] BOLES, W. W. & BOASHASH, B. A human identification technique using images of the iris and wavelet transform. *IEEE Transaction on Signal Processing*, vol. 46, no. 4, 1998.

- [8] CANNY, J. A computational approach to edge detection. *IEEE Transactions on PAMI-8*, no. 6, 1986.
- [9] CHEN, W. S. & YUAN, S. Y. A novel personal biometric authentication technique using human iris based on fractal dimension features. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 2003.
- [10] COOTES, T.; TAYLOR, C.; COOPER, D. & GRAHAM, J. Active shape models - their training and application. *Computer Vision and Image Understanding*, vol. 61, no. 1, January 1993.
- [11] COOTES, T.; TAYLOR, C.; HILL, A. & HALSAM, J. The use of active shape models for locating structures in medical images. *Proceedings of the 13th International Conference on Information Processing in Medical Imaging*, Springer-Verlag, 1993.
- [12] COOTES, T.; TAYLOR, C. & LANITIS, A. Active shape models: Evaluation of multi-mesolution method for improving image search. *Proceedings of the British Machine Vision Conference*, 1994.
- [13] CUI, J.; WANG, Y.; TAN, T.; MA, L. & SUN, Z. A fast and robust iris localization method based on texture segmentation. *Proceedings of the IEEE International Conference on Pattern Recognition*. <http://nlpr-web.ia.ac.cn/english/irds/papers/cuijl/SPIE.pdf>.
- [14] DARGHAM, J.; LIAM, L.; CHEKIMA, A. & FAN, L. Iris recognition using self-organizing neural network. *IEEE Student Conference on Research and Developing Systems*, 2002.
- [15] DAUGMAN, J. High confidence personal identification by rapid video analysis of iris texture. *Proceedings of the IEEE*.

- [16] DAUGMAN, J. High confidence visual recognition of person by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, 1993.
- [17] DAUGMAN, J. Iris recognition. *American Scientist*, vol. 89, no. 1, August 2001.
- [18] DAUGMAN, J. How iris recognition works. *Proceedings of 2002 International Conference on Image Processing*, vol. 1, no. 1, 2002.
- [19] DAUGMAN, J. New methods in iris recognition. *IEEE Transactions on system, man and cybernetics*, vol. 37, no. 5, 2007.
- [20] DAVIS, L. Handbook of genetic algorithms. *Van Nostrand Reinhold*, 1991.
- [21] DE CASTRO, F. C. Localização, segmentação e classificação automáticas de regiões de interesse para a determinação de maturidade óssea utilizando o método de tanner-whitehouse. Dissertação de mestrado, Universidade Federal de Uberlândia, Dezembro 2009.
- [22] DEMPSTER, A. P.; LAIRD, N. & RUBIN, D. Maximum likelyhood from incomplete data via the em algorithm. *Journal of the Royal Statistic Society*, vol. 39, no. 1, 1977.
- [23] DEY, S. & SAMANTA, D. Improved feature processing for iris biometric authentication system. *International Journal of Electrical and Electronics Engineering*, vol. 2, no. 2, 2007 Spring.
- [24] DU, Y.; IVES, R.; CHANG, C. I.; ETTER, D. & WELCH, T. Information divergence-based iris pattern recognition for automatic human identification. *Proceedings SPIE 5404*, 2004.

- [25] DU, Y.; IVES, R.; ETTER, D.; WELCH, T. & CHANG, C. A new approach to iris pattern recognition. *SPIE European Symposium on Optics/Photonics in Defense and Security*, 2004.
- [26] ERICSON, C. & ORDONEZ-REINOSO, I. Dialogue on uniform crossover. *GA-Digest*, vol. 5, no. 33, October 1991.
- [27] ESHELMAN, L. J. Bit-climbers and naive evolution. *GA-Digest*, vol. 5, no. 39, December 1991.
- [28] ESHELMAN, L. J. & SCHAFFER, J. D. Ga and very fast simulated re-annealing. *GA-Digest*, vol. 5, no. 33, Dezembro 1991.
- [29] F. W. GLOVER, G. A. K. Handbook of metaheuristics. *International Series in Operations Research & Management Science*, 2003.
- [30] FIELD, D. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optica Society of America*, 1987.
- [31] FISKEER, R. Making deformable template models operational. Dissertation de Mestrado, Technical University of Denmark, Lyngby, 2000.
- [32] FLOM, L. & SAFIR, A. Iris recognition system. *Patente USA 4 641 349*, 1987.
- [33] GOLDBERG, D. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.
- [34] GONZALEZ, R. & WOODS, R. E. *Digital Image Processing*. Addison-Wesley Publishing Company, Inc., 1992.

- [35] HAMARNEH, G. Active shape models, modeling shape variations and gray level information and an application to image search and classification. Relatório Técnico. R005/1998 (S2-IAG-98-1), Department of Signals and Systems, School of Electrical and Computer Engineering, Chalmers University of Technology, Goteborg, Sweden, September 1998.
- [36] HE, X. & SHI, P. A new segmentation approach for iris recognition based on hand-held capture device. *Pattern Recognition*, vol. 40, no. 4, 2007. DOI: 10.1016/j.patcog.2006.08.009.
- [37] HUANG, J.; WANG, Y.; TAN, T. & CUI, J. A new iris segmentation method for recognition. *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 3, no., 2004. DOI: 10.1109/ICPR.2004.1334589.
- [38] JAMUNDÁ, T. Reconhecimento de formas: a transformada de hough. *Seminário Visão Computacional - CPGCC/UFSC*, 2000.
- [39] KIM, J.; CHO, S. & CHOI, J. Iris recognition using wavelet features. *In Kluwer Academic Publishers, Journal of VLSI Signal Processing*, vol. 38, no. 2, 2004.
- [40] KONG, W. & ZHANG., D. Accurate iris segmentation based on novel reflection and eyelash detection model. *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, 2001.
- [41] KOVESI, P. Matlab function for computer vision and image analysis. www.cs.uwa.edu.au/~pk/Research/MatlabFns/index.html.

- [42] LANITIS, A.; TAYLOR, C. & COOTES, T. Automatic interpretation and coding of face images using flexible modes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, July 1997.
- [43] LIM, S.; LEE, K.; BYEON, O. & KIM, T. Efficient iris recognition through improvement of feature vector and classifier. *ETRI JOURNAL*, vol. 23, no. 2, 2001.
- [44] MA, L.; TAN, T. & WANG, Y. Iris recognition using circular symmetric filters. *Proceedings of the 16th International Conference on Pattern Recognition*, vol. 2, no. 1, 2002.
- [45] MA, L.; TAN, T.; WANG, Y. & ZHANG, D. Personal identification based on iris texture analysis. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, 2003.
- [46] MA, L.; TAN, T.; WANG, Y. & ZHANG, D. Efficient iris recognition by characterizing key local variations. *IEEE Transactions on Image Processing*, vol. 13, no. 6, 2004.
- [47] MA, L.; TAN, T.; ZHANG, D. & WANG, Y. Local intensity variation analysis for iris recognition. *Pattern Recognition*, vol. 37, no. 6, 2004.
- [48] MA, L.; WANG, Y. & TAN, T. Iris recognition based on multichannel gabor filtering. *Proceedings of the 15th Asian Conference on Computer Vision*, vol. 1, no. 1, 2002.
- [49] MARDIA, K.; KENT, J. & BIBBY, J. Multivariate analysis. *Academic Press*, 1995.
- [50] MASEK, L. Recognition of human iris patterns for biometric identification. Dissertação de Mestrado, The University of Western Australia, 2003.

- [51] MATAS, J.; GALAMBOS, C. & KITTLER, J. Progressive probabilistic hough transform. *British Machine Vision Conference*.
- [52] MIRA, J. & MAYER, J. Image feature extration for application of biometric identification of iris - a morphological approach. *IEEE Proceeding of the XVI Brazilian Symposium on Computer Graphics and Image Processings*, 2003.
- [53] MITCHELL, M. *An Introduction to Genetic Algorithms*. Massachusets: MIT Press, 1996.
- [54] MOSCATO, P. Np optimization problems, approximability and evolutionary computation: From practice to theory. Ph.d. dissertation, Universidade Estadual de Campinas, 2001.
- [55] NOH, S.; PAE, K.; LEE, C. & KIM, J. Multiresolution independent component analysis for iris identification. *The 2002 International Technical Conference on Circuits/Systems, Computers and Communications*, 2002.
- [56] OPPENHEIM, A. & LIM, J. The importance of phase in signals. *Proceedings of the IEEE*, vol. 69, no., 1981.
- [57] OTSU, N. A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man and Cybernetics*, vol. 9, no. 1, 1979.
- [58] PACHECO, M. A. C. Algoritmos genéticos: princípios e aplicações. *Laboratório de Inteligência Computacional Aplicada - ICA*.
- [59] PEREIRA, M. B. Uma proposta para o aumento da confiabilidade de um sistema de reconhecimento de íris e sua implementação através de algoritmo genético. Dissertação de mestrado, Universidade Federal de Uberlândia, Julho 2005.

- [60] PEREIRA, M. B. & VEIGA, A. C. P. Application of genetic algorithms to improve the reliability of an iris recognition system. *IEEE International Workshop on Machine Learning for Signal Processing - MLSP*, 2005.
- [61] PEREIRA, M. B. & VEIGA, A. C. P. A method for improving the reliability of an iris recognition system. *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing - PacRim*, 2005.
- [62] POURSAHERI, A. & ARAABI, B. N. Iris recognition for partially occluded images: Methodology and sensitivity analysis. *EURASIP Journal on Advances in Signal Processing*, 2007. Article ID 36751, DOI:10.1155/2007/36751.
- [63] PROENÇA, H. & ALEXANDRE, L. A. Ubiris: a noisy iris image database. *ICIAP 2005, 13th International Conference on Image Analysis and Processing*, vol. 3617 of Lecture Notes in Computer Science., no., 2005.
- [64] PROENÇA, H. & ALEXANDRE, L. A. Iris segmentation methodology for noncooperative iris recognition. *IEEE Proceedings Vision, Image, and Signal Processing*, vol. 153, no. 2, 2006.
- [65] ROY, K. & BHATTACHARYA, P. Iris recognition based on collarette region and asymmetrical support vector machine. *International Conference on Image Analysis and Recognition*. Springer Lecture Note Series in Computer Science (LNCS).
- [66] SCHAFFER, J. D.; CARUNA, R. A.; ESHELMAN, L. J. & DAS, R. A study of control parameters affecting online performance of genetic algorithms for function optimization. *Proceedings of the Third International Conference on Genetic Algorithms*, 1989.

- [67] SCHUCKERS, S. A. C.; SCHMID, N. A.; ABHYANKAR, A.; DORAIRAJ, V.; BOYCE, C. K. & HORNAK, L. A. On techniques for angle compensation in nonideal iris recognition. *IEEE Trans. SMC-B*, vol. 37, no. 5, 2007.
- [68] SIVARAMAN, A. Iris segmentation using daugman's integrodifferential operator. <http://www.mathworks.com/matlabcentral/fileexchange/15652>.
- [69] SMITH, J. M. *Evolutionary Genetics*. Oxford University Press, 1989.
- [70] SPEARS, W. M. Crossover or mutation? *Foundations of Genetics Algorithms*, vol. 2, no. 1, 1993.
- [71] STRANG, G. *Linear Algebra and its Applications*. Saunders, 1988.
- [72] T. COOTES, C. J. T. Statistical models of appearance for computer vision. Dissertação de Mestrado, University of Manchester., 2001.
- [73] T. JUNIOR, G. J. Algoritmos meméticos aplicados ao problema de nowait flowshop. Dissertação de mestrado, Universidade Estadual de Campinas, 2001.
- [74] TISSE, C.; MARTIN, L.; TORRES, L. & ROBERT, M. Person identification technique using human iris recognition. *International Conference on Vision Interface*, 2002.
- [75] VASTA, M.; SINGH, R. & NOORE, A. Reducing the false rejection rate of iris recognition using textural and topological features. *International Journal of Signal Processing*, vol. 2, no. 2, 2005.
- [76] VIGLIAZZI, D. *Biometria - Medidas de Segurança*. Visual Books, 2006.

- [77] WATSON, E. C. Desenvolvimento de um sistema biométrico para a identificação de padrões da íris. Dissertação de mestrado, Universidade do Rio de Janeiro, 2005.
- [78] WILDES, R. P. Iris recognition: an emerging biometric technology. *Proceedings of the IEEE*, vol. 85, no. 9, Setembro 1997.
- [79] WILDES, R. P.; ASMUTH, J.; GREEN, G.; HSU, S.; KOLCZYNSKI, R.; MATEY, J. & MCBRIDE, S. A system for automated iris recognition. *Proceedings of the IEEE Workshop on Applications of Computer Vision*, 1994.
- [80] ZHU, Y.; TAN, T. & WANG, Y. Biometric personal identification based on iris patterns. *Proceedings of the IEEE International Conference on Pattern Recognition*, vol. 2, no. 1, 2000.

Apêndice A

Introdução a Algoritmos Genéticos e Meméticos

A.1 Algoritmos Genéticos

Algoritmos genéticos (AG) são métodos utilizados para resolver problemas de busca e otimização. Eles são inspirados no princípio Darwiniano da evolução das espécies e no processo genético dos organismos biológicos. Atualmente, os AG estão sendo amplamente utilizados em aplicações científicas, comerciais e de engenharia. O grande crescimento do número de aplicações se deve principalmente ao fato de que os AG são algoritmos computacionalmente simples e, ao mesmo tempo, poderosos na tarefa de buscar otimizações. Além disso, eles não são limitados a restrições do espaço de busca como tamanho, descontinuidades, multi-modalidades, etc [33].

Charles Darwin estabeleceu que, no processo biológico, depois de muitas gerações, a população natural evolui de acordo com os princípios de seleção natural e da sobrevivência dos mais aptos. Na natureza, cada indivíduo possui seu próprio código genético que representa determinadas características desse indivíduo, como força, tamanho, etc. Quando dois indivíduos se acasalam, eles combinam seus códigos genéticos através de um processo de cruzamento, originando descendentes com novos códigos genéticos, que

podem torná-los tanto mais, quanto menos adaptados ao meio do que seus pais.

A Teoria da Evolução estabelece que, se um indivíduo é menos adaptado ao meio, ele terá poucas chances de transmitir seu código genético para as próximas gerações uma vez que dificilmente ele conseguirá acasalar-se. Assim, depois de muitas gerações, somente as características genéticas dos indivíduos mais adaptados continuarão a existir na população.

Eventualmente, durante um acasalamento, pode ocorrer uma mutação, ou seja, uma alteração no código genético que produz novas características que seriam improváveis de aparecer pela simples recombinação dos genes existentes. Se essas novas características forem ruins, isto é, não melhorarem a adaptação do indivíduo, elas serão brevemente extintas da população. Por outro lado, se elas melhorarem a adaptação do indivíduo, serão transmitidas às futuras gerações e continuarão existindo. Estes são os conceitos que os Algoritmos Genéticos (AG) tentam imitar com o objetivo de “evoluir” soluções para diversos problemas.

Cada possível solução para o problema é representada por um conjunto de parâmetros ou genes que são unidos para formar uma string de valores ou um cromossomo. A representação das possíveis soluções do espaço de busca define a estrutura do cromossomo a ser manipulado pelo AG. A representação mais comum é a binária. Ela é simples, fácil de ser manipulada pelos operadores genéticos e fácil de ser transformada em números inteiros ou reais [58].

No projeto de um AG é necessário a definição de uma *função de aptidão* que irá rotular um determinado cromossomo com um valor de aptidão proporcional à utilidade ou habilidade do indivíduo representado pelo cromossomo em questão. Para certos tipos de problema, como por exemplo, a maximização de uma função, o valor de aptidão é o próprio cálculo da função para um

determinado parâmetro. Porém, existem casos em que se pretende medir o desempenho de vários parâmetros ao mesmo tempo ou até da combinação entre eles.

O pseudo-código [A.1.1](#) contém os principais passos para a implementação computacional dos algoritmos genéticos. Primeiramente, na linha 3, uma população de indivíduos é gerada aleatoriamente e armazenada na variável *Pop*. Cada indivíduo é representado por um cromossomo que é constituído por um código que representa uma possível solução para o problema. Todos os indivíduos gerados devem ser avaliados e um valor de aptidão é associado a cada um deles de forma a representar o quanto cada solução é adequada ao problema em questão, ou, em outras palavras, o quanto os indivíduos estão adaptados ao meio. Esta avaliação é realizada na linha 5.

As linhas 6 até 19 mostram os passos para a criação da população da próxima geração. O primeiro passo é a implementação do chamado “*Elitismo*” que é utilizado para aumentar a convergência do AG. Este recurso consiste em preservar os n melhores indivíduos de cada geração. Estes bons indivíduos poderiam ser perdidos se eles não fossem selecionados para reprodução ou se eles fossem destruídos pelos operadores de cruzamento e mutação [53]. Portanto, os n melhores indivíduos são conduzidos diretamente da população atual para a população da próxima geração.

Para completar a nova população, os indivíduos são selecionados para reprodução. O mecanismo de seleção leva em consideração a aptidão dos indivíduos de forma que a probabilidade de um indivíduo ser selecionado para reprodução seja proporcional a seu valor de aptidão. Assim, se f_i é a aptidão do indivíduo i na população corrente, a probabilidade p_i do indivíduo i ser selecionado é proporcional a

```

1  // Procedimento AG
2
3  Pop = GeraPopulacaoInicial(TamanhoPop)
4  REPETIR
5      CalculaAptidao(Pop);
6      Elitismo(n, Pop, NovaPop);
7      i = n+1;
8      REPETIR
9          Ind1 = Selecao(Pop);
10         Ind2 = Selecao(Pop);
11         NovaPop(i) = Ind1;
12         NovaPop(i+1) = Ind2;
13         Se NoAleatorio1 < ProbCruzamento
14             [NovaPop(i), NovaPop(i+1)] = Cruzamento(Ind1, Ind2);
15         Se NoAleatorio2 < ProbMutacao
16             NovaPop(i) = Mutacao(NovaPop(i));
17             NovaPop(i+1) = Mutacao(NovaPop(i+1));
18         i = i+2;
19     ATÉ que NovaPop esteja completa
20     Pop = NovaPop;
21 ATÉ que o critério de parada seja satisfeito

```

Código A.1.1: *Procedimento do Algoritmo Genético.*

$$p_i = \frac{f_i}{\sum_{j=1}^{TamanhoPop} f_j} \quad (A.1)$$

onde *TamanhoPop* é o número de indivíduos na população.

Uma vez que dois indivíduos são selecionados (linhas 9 e 10), existe uma probabilidade “*ProbCruzamento*”, previamente definida, de seus genes serem recombinados pelo operador de cruzamento. Na aplicação desse operador, os dois indivíduos selecionados têm seus cromossomos “cortados” em uma posição aleatória e os genes localizados depois do corte são trocados entre os indivíduos para produzir dois novos indivíduos. Este procedimento está ilustrado na Figura A.1 e representa a forma mais simples do cruzamento, chamado cruzamento de um ponto de corte (*one-point crossover*). Os descendentes serão diferentes de seus pais, mas com características genéticas de

	Ponto de corte aleatório									
Cromossomo 1:	1	0	0	1	1	1	0	1	0	0
Cromossomo 2:	0	0	1	0	0	1	0	1	1	1
Descendente 1:	1	0	0	1	1	1	0	1	1	1
Descendente 2:	0	0	1	0	0	1	0	1	0	0

Figura A.1: *Representação do cruzamento de um ponto de corte*

ambos os genitores.

Para que o cruzamento aconteça com a probabilidade desejada, o seguinte procedimento é executado: um número aleatório “*NoAleatorio1*” entre 0 e 1 é gerado e se esse número for menor do que “*ProbCruzamento*”, o cruzamento é realizado e os dois novos indivíduos criados são adicionados à nova população; caso contrário, os dois indivíduos selecionados são levados à nova população sem sofrer nenhuma modificação (linhas 11 e 12).

O último operador genético é a mutação, que é aplicada com uma pequena probabilidade “*ProbMutacao*” (linhas 15 a 17). Esse operador é aplicado a cada descendente individualmente, geralmente depois do cruzamento, e altera o valor de um ou mais genes do cromossomo, que são escolhidos aleatoriamente. A Figura A.2 mostra o quinto gene do cromossomo sofrendo uma mutação simples. A mutação aumenta a diversidade da população e assegura que qualquer ponto do espaço de busca pode ser alcançado.

Quando a nova população está completa, todo processo descrito é repetido

	Gene escolhido aleatoriamente									
	↓									
Cromossomo:	1	0	0	1	1	1	0	1	0	0
Descendente:	1	0	0	1	0	1	0	1	0	0

Figura A.2: *Representação da mutação simples.*

até que o critério de parada seja satisfeito. O critério de parada pode ser o número de gerações, o tempo de execução ou um valor aceitável de aptidão.

O fluxograma da Figura A.3 mostra os passos para a execução de um AG de forma simplificada.

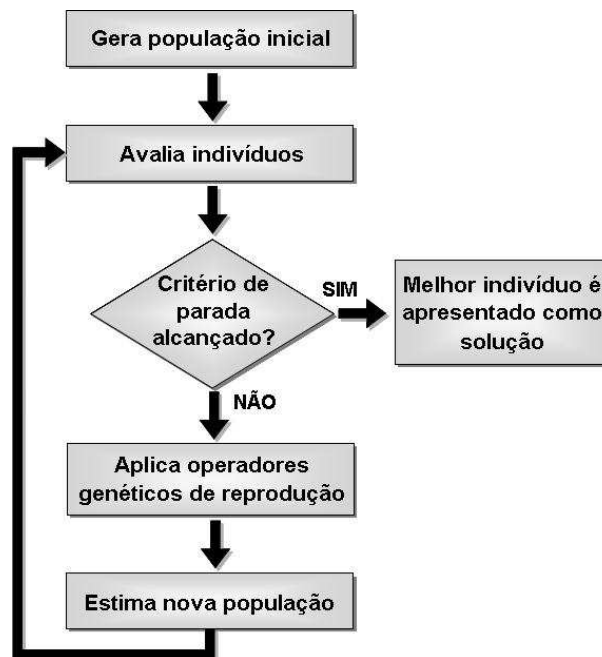


Figura A.3: *Fluxograma de um algoritmo genético genérico.*

É importante saber que os AGs não garantem encontrar a melhor de todas as soluções possíveis para um problema, mas geralmente eles são muito bons em encontrar uma solução aceitável em um tempo também aceitável.

A.2 Algoritmos Meméticos

Os chamados Algoritmos Meméticos (AM) combinam o conceito de “busca local” com os outros operadores tipicamente utilizados pelos Algoritmos Genéticos (AG). A busca local é a característica de um AM que o diferencia de um AG tradicional. Por causa da busca local, os algoritmos meméticos também são chamados “algoritmos genéticos híbridos” e dão origem ao conceito de

“evolução cultural”. Algoritmos meméticos levam em consideração não apenas a evolução genética dos indivíduos, mas também uma forma de evolução cultural. Na natureza, um certo indivíduo pode não ter sido gerado com uma boa aptidão, mas, as habilidades e experiências adquiridas ao longo de sua vida podem tê-lo tornado mais bem adaptado ao meio.

Em AM, depois que um indivíduo é gerado pelos operadores genéticos, ele passa pelo processo de evolução cultural. Assim, quando os indivíduos forem reproduzir, os descendentes irão herdar tanto características genéticas quanto culturais de seus pais.

O processo de evolução cultural é, geralmente, realizado por um algoritmo de busca local que irá definir os movimentos permitidos dentro do espaço de busca das soluções [54]. Um indivíduo adquire o máximo grau de conhecimento quando o algoritmo de busca local encontra um ótimo local dentro de seu espaço de busca [73]. Neste sentido, pode-se entender que os AM, diferentemente dos AG, procuram explorar todo conhecimento disponível a respeito do problema a ser estudado, com o objetivo de ajudar o processo evolutivo das soluções.

Um detalhe importante é que a busca local não é um processamento opcional nos AM, mas uma característica fundamental dessa metaheurística [29].

O pseudo-código A.2.1 mostra os passos de processamento de um AM. Pode-se verificar que a única diferença entre esse procedimento e o do AG é pela presença da busca local (linha 20) que é executada depois que o indivíduo é gerado pelos operadores de reprodução. O algoritmo de busca local avalia os indivíduos contidos em uma certa vizinhança definida pelo mecanismo de geração de vizinhança e, se algum indivíduo mais apto é encontrado, ele substitui o indivíduo original e é adicionado à nova população que está sendo formada.

```

1  // Procedimento AM
2
3  Pop = GeraPopulacaoInicial(TamanhoPop)
4  REPETIR
5      CalculaAptidao(Pop);
6      Elitismo(n, Pop, NovaPop);
7      i = n+1;
8      REPETIR
9          Ind1 = Selecao(Pop);
10         Ind2 = Selecao(Pop);
11         NovaPop(i) = Ind1;
12         NovaPop(i+1) = Ind2;
13         Se NoAleatorio1 < ProbCruzamento
14             [NovaPop(i), NovaPop(i+1)] = Cruzamento(Ind1, Ind2);
15         Se NoAleatorio2 < ProbMutacao
16             NovaPop(i) = Mutacao(NovaPop(i));
17             NovaPop(i+1) = Mutacao(NovaPop(i+1));
18         i = i+2;
19     ATÉ que NovaPop esteja completa
20     BuscaLocal(NovaPop);
21     Pop = NovaPop;
22 ATÉ que o critério de parada seja satisfeito

```

Código A.2.1: *Procedimento do Algoritmo Memético.*

O fluxograma da Figura A.4 mostra os passos para a execução de um AM de forma simplificada, onde é enfatizada a realização do procedimento de busca local.

A.3 Definição de Parâmetros para AG e AM

Como é possível perceber nas seções anteriores deste capítulo, a implementação de um AG ou de um AM requer a definição dos seguintes parâmetros: tamanho da população, número de gerações, taxa de cruzamento e taxa de mutação. Nesta seção, são realizados alguns comentários a respeito do efeito dos valores desses parâmetros no sistema e dos fatores que são levados em consideração no momento de definí-los.

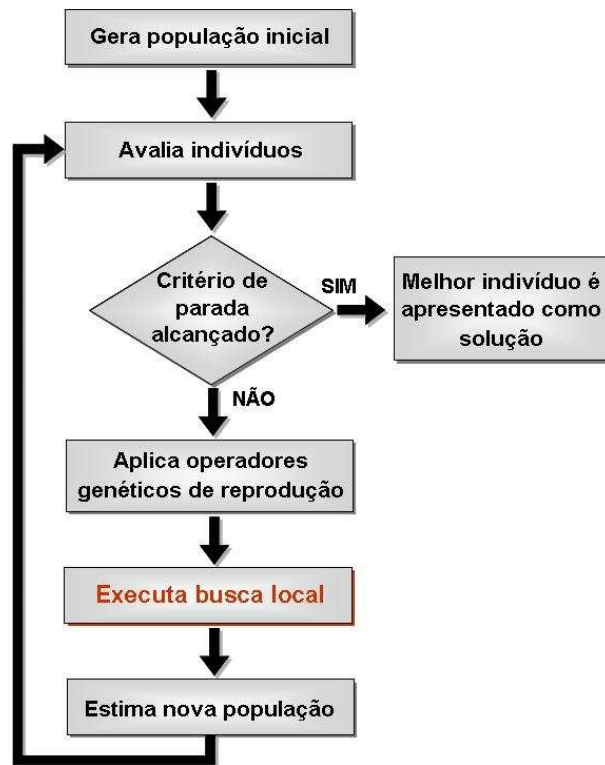


Figura A.4: *Fluxograma de um algoritmo memético genérico.*

O tamanho da população representa a quantidade de indivíduos existentes em uma determinada geração. Em geral, essa quantidade é definida e mantida fixa para todas as gerações processadas. Quanto maior é o tamanho da população, uma maior quantidade de possíveis soluções são representadas e maior a chance de bons indivíduos estarem presentes na população. Por outro lado, o aumento do tamanho da população resulta em um aumento do tempo de processamento de cada geração.

Para definir com relação a quantidade de gerações que são processadas, deve-se levar em consideração a seguinte análise. Se um algoritmo evolucionário é bem projetado, é esperado que, em uma determinada geração, os indivíduos estejam mais bem adaptados ao meio, ou, pelo menos igualmente adaptados, quando comparados aos indivíduos das gerações anteriores.

Portanto, quanto maior o número total de gerações executadas, maior a convergência do algoritmo para uma boa solução. Porém, é evidente que o tempo de processamento se torna maior, sendo importante estudar um meio termo adequado ao problema. O número de gerações pode ser usado como um critério de parada do AG ou AM.

A taxa de cruzamento representa a probabilidade de ocorrência de uma recombinação entre dois indivíduos selecionados. Se a taxa de cruzamento é 0%, toda a nova geração é composta de indivíduos que são exatamente a cópia de seus pais, pois, não é realizada nenhuma recombinação. Se a taxa for 100%, toda a descendência é gerada por cruzamento. Entretanto, é interessante que uma parte da população sobreviva até a próxima geração, preservando algumas características que podem ser importantes no futuro.

Neste trabalho, a taxa de mutação representa a probabilidade de se alterar um ou mais genes de um cromossomo. Se a taxa de mutação é 0%, nada é alterado e aumenta o risco do algoritmo cair em um ótimo local. Se a taxa é 100%, todos os indivíduos sofrem alguma alteração e o processo fica similar a uma busca aleatória.

Apêndice B

Método de Otsu para Limiarização

O método de Otsu [57] é utilizado para escolher automaticamente um ou mais limiares para separar uma imagem em classes de níveis de intensidade. Esta escolha é baseada inteiramente em cálculos realizados no histograma da imagem.

O método de Otsu assume que as funções de densidade de probabilidade (ou, os histogramas normalizados) dos níveis de intensidade de cada classe são funções gaussianas e busca maximizar a *variância entre classes*, uma medida bem conhecida utilizada na análise estatística discriminante. A idéia básica é que as classes com limiares bem estabelecidos devem ser distintas em relação aos valores de intensidade de seus pixels e, inversamente, que um limiar que oferece a melhor separação entre as classes em termos de valores de intensidade é o melhor limiar (limiar ótimo) [34].

O algoritmo do método de Otsu para escolher um único limiar ótimo e separar uma imagem em duas classes pode ser resumido nos seguintes passos:

1. Calcular o histograma normalizado da imagem de entrada e designar os componentes do histograma como p_i , $i = 0, 1, 2, \dots, L - 1$ que representa a probabilidade de ocorrência de cada nível de intensidade e pode ser

calculada pela Equação B.1.

$$p_i = \frac{n_i}{MN} \quad (\text{B.1})$$

onde n_i é o número de pixels da imagem com intensidade i , L é a quantidade de níveis de cinza na imagem, M é a quantidade de linhas da imagem e N é a quantidade de colunas da imagem.

2. Selecionar um limiar $T(k) = k$ ($0 < k < L - 1$) para estabelecer duas classes:

$$C_1 \rightarrow [0, k] \quad e \quad C_2 \rightarrow [k + 1, L - 1]$$

3. Calcular a probabilidade de que um pixel seja atribuído à classe C_1 ($P_1(k)$) e a probabilidade de que um pixel seja atribuído à classe C_2 ($P_2(k)$). Estas probabilidades são obtidas pelas somas acumuladas:

$$P_1(k) = \sum_{i=0}^k p_i \quad P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k)$$

4. Calcular as médias acumuladas $m(k)$ para $k = 0, 1, 2, \dots, L - 1$:

$$m(k) = \sum_{i=0}^k ip_i$$

5. Calcular a intensidade média global:

$$m_G = \sum_{i=0}^{L-1} ip_i$$

6. Calcular a variância entre as classes, para $k = 0, 1, 2, \dots, L - 1$:

$$\sigma_G^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$$

7. Obter o limiar de Otsu, k^* , como o valor de k para o qual $\sigma_G^2(k)$ é máxima:

$$\sigma_G^2(k^*) = \max(\sigma_G^2(k)) \quad 0 \leq K \leq L - 1$$

Se a máxima não é única, obter k^* pela média dos valores de k que correspondem aos diversos valores máximos detectados.

8. Obtido k^* , a imagem de entrada $f(x, y)$ é binarizada como:

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > k^* \\ 0 & \text{se } f(x, y) \leq k^* \end{cases}$$